

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Zavedení Business Intelligence pro zvýšení efektivity řízení IT služeb
ve výrobním podniku

Implementation of Business Intelligence for Increase of IT Services
Management Efficiency in Manufacturing Company

Student: Radek Vymětal

Vedoucí diplomové práce: Ing. Radek Němec, Ph.D.

Ostrava 2015

Zadání diplomové práce

Student:

Bc. Radek Vymětal

Studijní program:

N6209 Systémové inženýrství a informatika

Studijní obor:

6209T025 Systémové inženýrství a informatika

Téma:

Zavedení Business Intelligence pro zvýšení efektivity řízení IT služeb ve výrobním podniku

Implementation of Business Intelligence for Increase of IT Services Management Efficiency at Manufacturing Company

Zásady pro vypracování:

1. Úvod
2. Teoretická východiska implementace systému Business Intelligence
3. Analýza stavu řízení IT služeb
4. Zavedení Business Intelligence jako nástroje pro zvýšení efektivity řízení IT služeb
5. Závěr

Seznam použité literatury

Seznam zkratk

Prohlášení o využití výsledků diplomové práce

Seznam příloh

Přílohy

Seznam doporučené odborné literatury:

GOLFARELLI, Matteo and Stefano RIZZI. *Data warehouse design: modern principles and methodologies*. New York: McGraw-Hill, 2009. 500 p. ISBN 978-0-07-161039-1.

KIMBALL, Ralph and Margy ROSS. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. 3rd ed. Indianapolis: Wiley, 2013. 600 p. ISBN 978-1-118-53080-1.

NĚMEC, Radek. *Principy projektování a implementace systémů Business Intelligence*. Ostrava: VŠB-TU Ostrava, 2014. 159 s. ISBN 978-80-248-3452-8.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Radek Němec, Ph.D.**

Datum zadání: 21.11.2014

Datum odevzdání: 25.04.2015



doc. Ing. Jana Hančlová, CSc.
vedoucí katedry






prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Prohlášení

Prohlašuji, že svou diplomovou práci na téma „Zavedení Business Intelligence pro zvýšení efektivity řízení IT služeb ve výrobním podniku“ jsem vypracoval včetně všech příloh samostatně pod vedením mého vedoucího diplomové práce, odborného konzultanta Ing. Pavla Meci a s použitím dostupné odborné literatury a dalších informačních zdrojů, které jsem všechny uvedl v seznamu použité literatury.

V Ostravě dne 14. 7. 2015


.....
(podpis autora)

Poděkování

Rád bych na tomto místě poděkoval Ing. Radku Němcovi, Ph.D. za odborné vedení, náměty, rady a připomínky při psaní této diplomové práce. Současně bych rád poděkoval společnosti StoraEnso, konkrétně Ing. Pavlu Mecovi, který mi umožnil tuto práci realizovat a předal mi cenné praktické rady nezbytné k úspěšnému dokončení celé práce.

Obsah

1) ÚVOD.....	5
2) TEORETICKÁ VÝCHODISKA IMPLEMENTACE SYSTÉMU BUSINESS INTELLIGENCE. 6	
2.1) ÚVOD DO TERMINOLOGIE	6
2.2) POSTAVENÍ BUSINESS INTELLIGENCE V SYSTÉMU ŘÍZENÍ.....	7
2.2.1) Měření výkonnosti podnikových procesů.....	10
2.3) MULTIDIMENZIONALITA A MULTIDIMENZIONÁLNÍ POHLED NA DATA	11
2.3.1) Multidimenzionalita dat a prostředí relační databáze	12
2.3.2) Multidimenzionalita dat a prostředí OLAP databáze.....	13
2.4) ARCHITEKTURA A KOMPONENTY SYSTÉMU BI	14
2.4.1) Produkční a zdrojové systémy	15
2.4.2) Dočasné úložiště dat (Data Staging Area - DSA)	15
2.4.3) Operativní úložiště dat (Operation Data Store - ODS).....	16
2.4.4) Transformační nástroje (Extract, Transform, Load - ETL).....	16
2.4.5) Integrovaní nástroje (Enterprise Application Integration - EAI).....	16
2.4.6) Datová tržiště (Data Marty).....	16
2.4.7) Centrální datové úložiště.....	17
2.4.8) OLAP (On-line Analytical Processing)	20
2.4.9) Komponenty uživatelského rozhraní systému BI	20
2.4.10) Nástroje pro zajištění kvality dat.....	21
2.4.11) Nástroje pro správu metadat.....	21
2.5) METODIKA BUDOVÁNÍ SYSTÉMU BUSINESS INTELLIGENCE	22
2.5.1) Analýza požadavků.....	22
2.5.2) Dimenzionální modelování.....	23
2.5.3) Fyzický návrh datového modelu	37
2.5.4) Návrh a vývoj ETL procesů a úloh.....	39
3) ANALÝZA STAVU ŘEŠENÍ ŘÍZENÍ IT SLUŽEB.....	46
3.1) ÚVOD O SPOLEČNOSTI	46
3.2) ANALÝZA PODNIKOVÝCH SYSTÉMŮ.....	48
3.2.1) Zadání práce	48
3.2.2) Analýza podnikových procesů	49
4) NÁVRH ŘEŠENÍ PRO ZEFEKTIVNĚNÍ ŘÍZENÍ SLUŽEB POMOCÍ NÁSTROJŮ BUSINESS INTELLIGENCE.....	53
4.1) DIMENZIONÁLNÍ MODELOVÁNÍ.....	53
4.1.1) Výběr podnikového procesu	53
4.1.2) Deklarace podrobnosti informací o podnikovém procesu.....	58

4.1.3)	<i>Identifikace dimenzí</i>	59
4.1.4)	<i>Identifikace a výběr numerických faktů (metrik, ukazatelů)</i>	70
4.2)	FYZICKÝ NÁVRH.....	76
4.2.1)	<i>Agregační strategie</i>	76
4.2.2)	<i>Indexační strategie</i>	76
4.2.3)	<i>Rozdělení tabulek (partitioning)</i>	77
4.3)	NÁVRH A VÝVOJ PROCESŮ A ÚLOH DATOVÉ PUMPY (ETL)	77
4.3.1)	<i>Nakreslit více-úrovňový (High-level) plán</i>	78
4.3.2)	<i>Výběr ETL nástroje</i>	78
4.3.3)	<i>Vývoj výchozích strategií</i>	78
4.3.4)	<i>Detailní analýza navrhovaného datového skladu</i>	79
4.3.5)	<i>Plnění dimenzionálních tabulek historickými daty</i>	80
4.3.6)	<i>Plnění faktových tabulek historickými daty</i>	84
4.3.7)	<i>Plnění dimenzionálních tabulek přírůstkovými daty</i>	85
4.3.8)	<i>Plnění faktové tabulky přírůstkovými daty</i>	86
4.3.9)	<i>Agregace dat v tabulkách a nahrání dat do OLAP databáze</i>	87
4.3.10)	<i>Automatizace fungování ETL systému</i>	87
4.4)	TVORBA OLAP KOSTEK PRO TABULKY FAKTŮ	88
4.4.1)	<i>OLAP kostka pro tabulku faktů FactAccessDelivery</i>	88
4.4.2)	<i>OLAP kostka pro tabulku faktů FactHardwareDelivery</i>	88
4.4.3)	<i>OLAP kostka pro tabulku faktů FactRfc</i>	88
4.4.4)	<i>OLAP kostka pro tabulku faktů FactWorkforceStaffPlanning</i>	89
4.4.5)	<i>OPAP kostka pro tabulku faktů FactBudgetPlanning</i>	90
4.4.6)	<i>Modifikace datových kostek</i>	91
4.5)	REPORTING A TESTOVÁNÍ	91
4.5.1)	<i>Workforce Staff Planning</i>	92
4.5.2)	<i>Budget Planning</i>	94
4.5.3)	<i>Access Delivery</i>	95
4.5.4)	<i>Hardware Delivery</i>	96
4.5.5)	<i>Request for change</i>	97
5)	ZÁVĚR	101
	ZDROJE	103
	SEZNAM ZKRATEK	105
	SEZNAM PŘÍLOH	108

1) Úvod

Podniky jsou v současnosti vystavovány velkému tlaku na zvyšování konkurenceschopnosti, kterému se snaží čelit zaváděním nových moderních technologií, stejně tak jako efektivním využíváním informací, jak ve firmě, tak i na trhu. K tomu aby obstály, jsou nuceny měřit efektivitu svých činností. Do popředí se tak dostává efektivita, rychlá dostupnost, kvalita a rozsah získávaných informací.

Indicií k napsání této diplomové práce je současné masivnější uplatňování konceptu business intelligence, využitelného taktéž pro zvýšení efektivity monitorování a identifikaci příčin začínajících problémů v oblasti IT služeb.

Cílem diplomové práce je navrhnout řešení, které zefektivní řízení IT služeb v konkrétním výrobním podniku, se záměrem zvýšení vypovídající hodnoty dat, zavedení interaktivnosti reportingu a detailnější sledování plnění podnikových KPI pro IT služby.

První část práce je věnována teoretickému představení základních pojmů business intelligence, informačního systému, vztahu business intelligence k systému řízení. Dále se zaměřuje na principy a požadavky manažerů pro řešení business intelligence. Následně jsou uvedeny jednotlivé komponenty BI, kdy větší pozornost je věnována návrhu datového skladu. Největší důraz je upřen na budování systému BI, jehož hlavní části je dimenzionální modelování. Závěr teoretické části je věnován fyzickému modelování a návrhu procesů extrakce, transformace a nahrání dat.

V první kapitole praktické části bude představena společnost, pro kterou je zpracována diplomová práce. Následně bude provedena podrobná analýza jednotlivých procesů společnosti a výběr vhodných procesů k následnému vytvoření business intelligence řešení pro měření výkonnosti jednotlivých procesů organizace. V dalších kapitolách se budeme věnovat modelování zvolených procesů a implementaci řešení datového skladu. Na závěr práce bude proveden návrh datových kostek a možných reportů. Věřím, že mé výsledné řešení přinese společnosti lepší možnosti reportingu a bude využíváno v plné míře.

V práci bude využita analýza požadavků na základě interview a následně, přístup získávání požadavků za pomoci řízení zdroji dat. Dále bude uplatněn Kimballův přístup pro modelování a návrh dimenzionálního modelu, návrh nahrávacích rutin, vytvoření datových kostek a výsledných reportů v prostředí Microsoft Excel.

2) Teoretická východiska implementace systému Business Intelligence

2.1) Úvod do terminologie

Aby byly lépe pochopeny principy dnešního pojetí Business Intelligence (BI), bude zde být nejprve vysvětleno několik základních pojmů - data, informace, znalosti, systém a informační systém.

Data jsou zjednodušeně charakterizována jako libovolná posloupnost znaků. Pokud je využita úplná definice, pak data jsou představována jako „*jakékoli vyjádření (reprezentace) skutečnosti, které je schopné přenosu, interpretace či zpracování. Účelem dat je přenášet a dále zpracovávat odraz skutečnosti. Jsou to jakékoli zaznamenané poznatky či fakta*“ (Kalousková, 2014, str. 1).

Informace jsou chápány v obecném smyslu jako sdělování zpráv, poznatků, událostí nebo jevů. Podle filozofického hlediska je brána informace jako nehmotný statek, který je představován tím, co je vnímáno člověkem a význam je získáván až v určitém kontextu. Uvedeme si ještě třetí vymezení tohoto pojmu vycházející z teorie informace, kde se bere informace jako zpráva upřesňující zřejmá fakta o jevech nebo objektech reálného světa. Její množství je znázorňováno mírou neurčitosti, která je odstraněna zprávou (Tvrdíková, 2008).

Znalosti jsou podle definice Roberta M. Hayese „*výsledkem porozumění informací, která byla právě sdělena, a její integrace s dřívějšími informacemi. Znalosti lze charakterizovat také jako informace o tom, jak využít jiné informace a data (a to i ve vzájemných kombinacích)*“ (Šlapák, 2003, str. 4).

Systém definujeme jako účelovou množinu prvků a vazeb mezi nimi. Zmíněný pojem může být využíván pro označení části reálného světa se specifickými vlastnostmi (Tvrdíková, 2008).

Informační systém, je všeobecně charakterizován „*jako soubor lidí, metod a technických prostředků zajišťujících sběr, přenos, uchovávání, zpracování a prezentaci dat s cílem tvorby a poskytování informací dle potřeb příjemců informací činných v systémech řízení*“ (Tvrdíková, 2000, str. 18).

Podnikatelské prostředí dnešní doby je vyznačováno vysokou proměnlivostí v oblastech vědeckotechnického rozvoje, ekonomických a politických změn, čemuž by mělo být přizpůsobováno i chování podniku. Proto je používán systém řízení, kterým jsou využívána data získaná pomocí informačních systémů. Výhodou tohoto propojení je rychlá odezva na

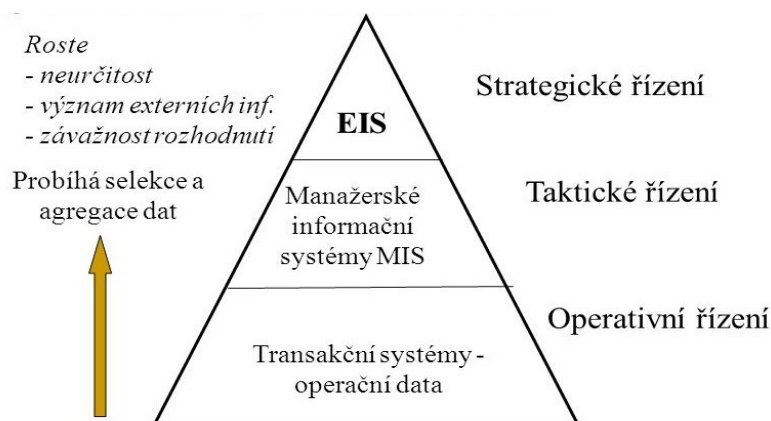
základě nových dat poskytnutých informačními systémy sloužících pro účinnou adaptaci a úpravu prováděných změn (Tvrdíková, 2008).

2.2) Postavení Business Intelligence v systému řízení

V organizacích nacházíme dvě rozdílné informační potřeby, jedna vyžadující operativní úroveň rozhodování, jako řešení stížností od zákazníků, získávání nových zákazníků, vyřizování objednávek zkrátka konkrétní kroky, které jsou činěny a realizovány v daný čas na daném místě. Pro tuto potřebu jsou běžně využívány pouze aktuální údaje, a tak pracujeme jen s několika záznamy současně. Tato oblast je zastřešována transakčními systémy, zpracování je pak prováděno systémem transakčního zpracování známého jako „On-line Transaction Processing“ (OLTP).

Druhou potřebou chápeme dlouhodobé a strategické rozhodování, které je nejčastěji podloženo jak historickými údaji, tak i externími zdroji z okolí podniku. Ty jsou využívány pro podrobnou analýzu agregovaných dat, pomocí kterých jsou sdružovány stovky až tisíce záznamů. Pro tyto potřeby je nezbytné mít data organizována pro rychlou odezvu při dotazování a vysoký výkon. Dané nároky jsou používány systémy využívající „On-Line Analytical Processing“ (OLAP). Principu OLAP je použito v současnosti nejvíce v konceptu BI (Kimball a Ross, 2013).

Výše uvedené dvě potřeby na práci s informacemi byly rozděleny i podle úrovně řízení společnosti, jak je vidno na obrázku číslo 2-1 „Pohled na informace z hlediska úrovně řízení“. Za nejefektivnější je považováno, když jsou vzájemně propojeny oba dva typy těchto systémů, a tím je umožňováno postupné zpracování a předávání pro úplné využití informace na všech úrovních řízení.



Obr. 2-1 Pohled na informace z hlediska úrovně řízení
Zdroj: (Molnár, 2009)

Na nejspodnější úrovni této pyramidy se nachází **transakční systémy**, které se využívají pro operativní úroveň řízení. Pomocí transakčních systémů jsou zajištěny základní procesy v organizaci, například zpracování objednávek, plánování výroby a další. Podle třídění uvedeného v předchozí pasáži se jedná o první členění vyžadující operativní rozhodování. Na této a taktické úrovni se používají kromě základních provozních systémů i **manažerské informační systémy** („Management Information System“ (MIS)), které byly převzaty z ekonomických a účetních systémů (Tvrdíková, 2008). MIS je chápán v obecné koncepci jako souhrn počítačových systémů, sloužících pro specifické účely v systému řízení, kde jsou prováděny na pozadí procesy zpracování a ukládání transakčních dat (Němec, 2014).

Uprostřed pyramidy se nachází kromě MIS i **systémy pro podporu rozhodování** („Decision Support Systems” – (DSS)), které definujeme jako rozšiřitelnou sadu interaktivních IT technik a nástrojů, navrhnutých pro zpracování a analýzu dat k podpoře manažerů při rozhodování. Způsob práce těchto systémů je založen na sjednocení jednotlivých zdrojů manažerů s počítačovými zdroji pro zlepšení kvality rozhodnutí. V praktickém pojetí se jedná o IT systém, kterým je napomáháno manažerům činit rozhodnutí nebo umožňován výběr z množství různých alternativ. A v takovém případě je vypočítávána hodnota každé alternativy samostatným systémem (Golfrelli, 2009).

Vrchol řídicí pyramidy je pokrýván **systémy pro podporu vrcholového řízení (EIS)** neboli manažerské aplikace BI. Tyto aplikace jsou využívány především vrcholovým vedením organizace. Jsou navrženy pro zprostředkování přístupu k externím datům a zároveň byly napojeny na informační systémy společnosti. Z operativních dat jsou vytvářena agregovaná data s vysokou vypovídající hodnotou, umožňující vyhledávání zákonitostí, odchylek od plánů, práci s historií i anticipaci budoucího vývoje (Tvrdíková, 2008).

Ve strategickém řízení jsou užívány od roku 2005 nové aplikace zvané **expertní systémy (ES)**, pracující s konceptem umělé inteligence. Využívající lidských znalostí uložených v programech pro řešení velmi složitých problémů (Němec, 2014).

Nyní si uvedeme příklady z četných definic BI. Z výše uvedené definice EIS je odvozena v pořadí první z nich, podle níž BI představuje *„systém nástrojů, projektových řešení a organizačních opatření, umožňujících řízení organizací podle znalostí“* (Tvrdíková, 2008, str. 95). Autor práce je však nakloněn následující definici pro její lépe vypovídající význam, kdy BI je používán pro transformaci surových dat pocházejících ze systémů společnosti a různých externích datových zdrojů do informací, které obsahují přidanou hodnotu, umožňující efektivní rozhodování založené na faktech. K tomu je využíván sběr dat,

jejich konsolidace, organizování, ukládání, šíření, analýza a poskytování rychlého a snadného přístupu k datům (Ballard, 2006, str. 41, volně přeloženo).

Novotný (2005, str. 19) uvádí ve své publikaci následující definici: „*BI je sada procesů, know-how, aplikací a technologií, jejichž cílem je účinně a účelně podporovat řídicí aktivity ve firmě. Podporující analytické, plánovací a rozhodovací činnosti organizací na všech úrovních a ve všech oblastech podnikového řízení, tj. prodeje, nákupu, marketingu, finančního řízení, controllingu, majetku, řízení lidských zdrojů, výroby a dalších.*“

Na základě uvedených definic a vývoje můžeme shrnout principy BI do následujících bodů:

- BI je určena pro analytické a plánovací aplikace, a tomu je uzpůsobena organizace dat,
- data jsou uložena na potřebných úrovních detailu (granularity), tedy současně detailní i agregovaná,
- pomocí ní je umožňováno vyhodnocení podnikových ukazatelů z různých úhlů pohledu, tzn. je založena na multidimenzionalitě uložení a zpracování dat,
- je využíváno časové dimenze, kdy data jsou ukládána včetně časového hlediska,
- klade vyšší nároky na kvalitu dat.

Uvedené principy se odráží i v požadavcích manažerů na BI systém, které shrnují Kimball a Ross (2013) do následujícího seznamu:

- BI systémem musí být umožněn jednoduchý a rychlý přístup k informacím, kdy data musí být intuitivní a pochopitelná pro koncové uživatele,
- dodávat důvěryhodné a konzistentní informace získané z několika rozdílných zdrojů očištěné od chyb a se zajištěnou datovou kvalitou,
- mít výstižné a jednotné pojmenování jednotlivých atributů a přesný popis transformačních pravidel,
- schopnost uzpůsobit se měnícím se potřebám společnosti při zachování konzistence současných dat a využívaných aplikací,
- poskytovat informace včas, kdy velké množství nových dat musí být transformováno do přehledných konsolidovaných informací, a to co nejrychleji pro možnosti operativního rozhodování,
- zaručovat vysoký stupeň zabezpečení uchovávaných informací především proto, že v systému jsou obvykle obsažena důvěrná a utajovaná data společnosti,
- zajišťovat autoritativní a důvěryhodný zdroj dat pro podporu rozhodování, kdy nejdůležitější výstupy jsou založeny na analytickém důkazu,

- uživatelé se musí se systémem BI ztotožnit a přijmout jej, kdy pozornost je věnována tomu, aby uživatelé pochopili výhody nového systému, začali důvěřovat datům, jež jsou systémem vyprodukována, a aktivně systém využívali.

Pokud se vrátíme k nadpisu kapitoly postavení BI v systému řízení, podle Němce (2014) můžeme v dnešní době hovořit o tom, že BI již proniklo na všechny úrovně řízení – strategickou, taktickou i operativní. Význam pro strategickou úroveň řízení již byl demonstrován v předchozím textu. Na úrovni taktické se BI zaměřuje na plánování a řešení střednědobých cílů, realizaci dílčích plánů a monitorování podnikových procesů. Na operativní úrovni pomáhá řešit konkrétní každodenní činnosti v souvislosti s prováděním podnikových procesů.

2.2.1) Měření výkonnosti podnikových procesů

Celé řešení BI zde prezentované se zaměřuje na měření výkonnosti podnikové informatiky/podnikových procesů. Výkonnost podniku a její hodnocení jsou brány rozdílně pro různé skupiny lidí, kteří sledují odlišné cíle. Tyto skupiny jsou tvořeny vedoucími pracovníky, vlastníky, zákazníky nebo zaměstnanci firem (Ballard, 2006).

Řízení výkonnosti je definováno jako „*kombinace řízení, metodik a metrik podporovaných aplikacemi, nástroji a infrastrukturou, která umožňuje uživatelů definovat, monitorovat a optimalizovat výsledky a výstupy tak, aby bylo dosaženo cílů osobních či cílů organizační jednotky v souladu se strategickými cíli stanovenými na různých úrovních řízení podniku (osobní, procesní, skupinové a korporátní cíle podniku nebo podnikatelského ekosystému)*“ (Novotný, 2010, str. 20).

Řízení podnikové výkonnosti je spojováno s pojmem „**Corporate Performance Management**“ (CPM), který je udáván jako hlavní představitel systému řízení výkonnosti. Pojem je definován společností Gartner: „*CPM je koncept řízení, který popisuje všechny procesy, metodiky, metriky a systémy potřebné k měření a řízení výkonnosti organizace*“ (Novotný, 2010, str. 20). Definicí je řečeno, že CPM je tvořen jednotlivými podnikovými procesy, manažerskými metodami a metrikami.

Podnikový proces můžeme definovat dle normy ISO 9001 jako soubor vzájemně působících činností, kterými jsou přeměňovány vstupy na výstupy. Jinými slovy rozumíme podnikovému procesu například jako sérii logicky spojených aktivit nebo úkolů (jako plánování, produkce nebo prodej) společně vykonávaných za účelem vyprodukování výsledků (Business dictionary, 2015, volný překlad).

Metodologický základ podnikového řízení je tvořen **manažerskými metodami** a jsou do nich zahrnovány Balanced Scorecard, Six Sigma, aktivita Based Costing a další.

Práce je zaměřována především na **metriky**, které jsou postaveny na klíčových ukazatelích ve vztahu k dimenzím s potřebnými charakteristikami – vnitřní struktura, kalkulace, zdroje, data a podobně. Zmíněné metriky jsou přiřazovány k jednotlivým podnikovým procesům a v rámci CPM jsou rozdělovány na tři skupiny.

Klíčové indikátory výsledků (KRI) se zaměřují na náklady, výnosy a spokojenost zákazníků. **Klíčové indikátory výkonnosti (KPI)** jsou používány pro měření výkonnosti lidských zdrojů, interních procesů. **Ostatní indikátory výkonnosti (PI)** jsou vymezovány jako specifické ukazatele výkonnosti vybraných procesů, zdrojů či pracovních týmů.

CPM je postaven na principech BI, koncept je ovšem rozšiřován o řízení, plánování a podnikovou strategii. Je úzce spojen s metodami a metodikami pro řízení výkonnosti. Jeho aplikace je soustředěna na splnění celopodnikových i dílčích cílů napříč jednotlivými odděleními podniku (Novotný, 2010).

2.3) Multidimenzionalita a multidimenzionální pohled na data

Požadavek pohledů uživatelů na sledovaná data z více hledisek a jejich kombinace jsou řešeny principem multidimenzionality (Pour, 2012). Multidimenzionální pohled na data je determinován třemi faktory - dimenze, metrika a čas.

Dimenze jsou představovány jednotlivými úhly pohledu na data. Prostřednictvím dimenze je i zobrazováno analytické hledisko pro hodnocení sledovaných ukazatelů. Z informatického aspektu se dimenze jeví jako struktura dat (databázová tabulka), kdy prvky dimenzí jsou vesměs uspořádány v hierarchické struktuře, například rozdělením na kategorie. Za měřitelné ukazatele považujeme **metriky**, které prezentujeme z různých úhlů pohledu. **Čas** je chápán za nepostradatelný pro zobrazení vývoje metrik (Němec, 2014).

Principem multidimenzionality jsou přinášeny specifické nároky na organizaci dat v databázi. Především uspořádání v logických shlucích podle účelu jejich použití, tedy okolo konkrétních faktů, které jsou sledovány uživatelem. Dané požadavky jsou řešeny pomocí dvou přístupů, a to multidimenzionalitou dat a prostředím relační databáze nebo pomocí multidimenzionality dat a prostředí OLAP databáze (Pour, 2012).

2.3.1) Multidimenzionalita dat a prostředí relační databáze

Databáze je založena na soustavě tabulek dimenzí propojených s tabulkou faktů. Tabulka faktů je spojena pomocí cizích klíčů s dimenzionálními tabulkami, které jsou používány k uchovávání popisných informací o sledovaných ekonomických ukazatelích uložených v tabulce faktů. Blíže budou obě tabulky vysvětleny v následující části.

Faktové tabulky jsou užívány pro ukládání údajů o výkonnosti podnikových procesů společnosti. Takovými údaji je myšleno například množství a cena prodaného zboží. Nejčastěji se využívá numerických dat, v jedné faktové tabulce je možno nalézt jen údaje stejné granularity neboli stejné úrovně detailu. Často jsou v nich obsaženy tisíce, miliony či až miliardy záznamů. Primární klíč faktových tabulek je obvykle tvořen složením z cizích klíčů spojujících faktovou tabulku s tabulkami dimenzí.

Pro ukládání textových dat, které se v čase nemění, jsou naopak upřednostňovány tabulky **dimenzí**. Je v nich popisováno „kdo, co, kde, kdy, jak a proč“, jsou spojeny s událostmi - popisují data z tabulky faktů. Často zde nalézáme mnoho sloupců nebo atributů. Ale na rozdíl od faktových tabulek v nich není obsaženo tolik řádků (jednotlivých záznamů). Každá dimenzionální tabulka je identifikována pomocí primárního klíče, propojeného s faktovou tabulkou.

Dimenzionální schéma je vytvořeno spojením výše uvedených tabulek faktů a dimenzí. Charakteristický znak schématu je vyznačován jednoduchostí a symetrií. Jsou používána dvě základní schémata, a to „hvězdy“ (Star) a „sněhové vločky“ (Snowflake) (Golfarelli, 2009).

Ve schématu hvězdy je obsažena celá hierarchická struktura dimenze v rámci jedné tabulky faktů. Tudiž identifikátory a popisné údaje jsou opakovány ve vyšších úrovních. Výhodu daného řešení vidíme v tom, že je minimalizován počet vzájemných propojení tabulek při dotazování (join operace), což je vedeno k vyšší rychlosti zpracování. Naopak nevýhodu můžeme spatřovat v obsáhlosti dimenzí a ve vysoké úrovni detailnosti dat v dimenzi. Při opakovanější aktualizaci záznamů jsou zaznamenány potíže i v dimenzionálních tabulkách. Vzniklé problémy jsou řešeny pomocí normalizace tabulek, kdy nově vzniklé schéma připomíná sněhovou vločku.

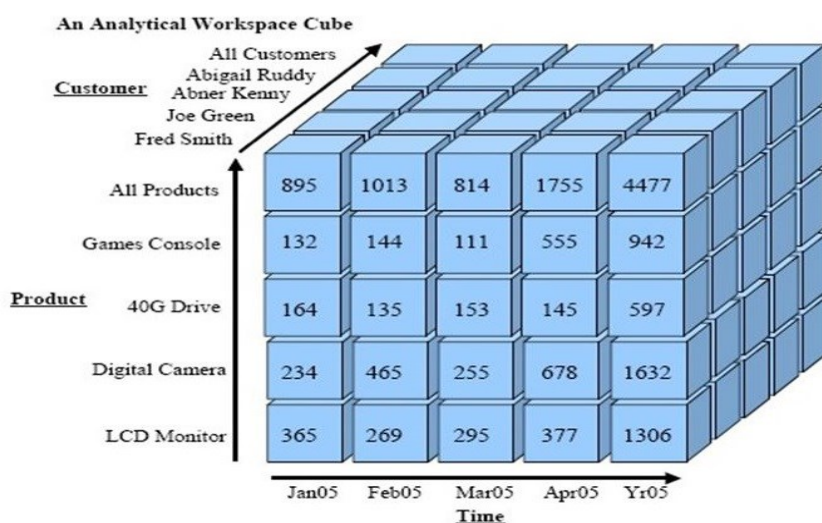
Sněhová vločka je charakterizována víceúrovňovým modelem tabulek (vyšším stupněm normalizace), hierarchie podle kardinality jsou vytvářeny dimenzí a jsou provázány pomocí sdílených dimenzí. Jsou doplňovány dalšími dimenzemi a číselníky. Toto řešení se vypořádává s problémy souvisejícími s návrhem rozsáhlých a často se měnících dimenzí. Současně je však

zvyšován počet vzájemných spojení mezi tabulkami, a s tím jsou spojeny možné problémy při velkých agregacích dat a ad-hoc dotazech (Pour, 2012), (Němec, 2014).

Na závěr této podkapitoly uvedeme hlavní výhody a nevýhody multidimenzionality dat v prostředí relační databáze. Hlavní nedostatek tohoto přístupu k uložení multidimenzionálních dat je nalézán v potřebě spojování velkých tabulek, což se odráží na výkonu. Částečné řešení je nacházeno v denormalizaci tabulek orientovaných na maximální výkon a zavedení redundance dat (Golfarelli, 2009).

2.3.2) Multidimenzionalita dat a prostředí OLAP databáze

Multidimenzionální databáze jsou optimalizovány pro interaktivní využívání uložených dat. Cílí na úzeji definované požadavky oproti relačním databázím, které jsou zaměřeny na širší oblasti činnosti organizace. Na tomto místě si uvedeme definici slova OLAP „*OLAP rozumíme informační technologii, založenou především na koncepci multidimenzionálních databází. Jejím hlavním principem je několikadimenzionální tabulka umožňující rychle a pružně měnit jednotlivé dimenze a měnit tak pohledy uživatele na modelovanou ekonomickou realitu*“ (Pour, 2012, str. 21).



Obr. 2-2: Princip multidimenzionální databáze na bázi OLAP
Zdroj: (Oracle Olap, 2009)

Obsah dimenzí je reprezentován složkami dimenzí a jejich agregací do jednoho bodu se vytváří prvek OLAP kostky. Prvky jsou pak zpravidla uspořádány v hierarchické struktuře. OLAP databáze jsou tvořeny uskupením souvisejících OLAP kostek (Novotný, 2005).

Snazší představa OLAP kostky je dána díky obrázku číslo 2-2 Princip multidimenzionální databáze na bázi OLAP, kdy tržbami, typem výrobku a časem jsou představovány jednotlivé dimenze.

Na závěr si uvedeme opět výhody a nevýhody, kdy za hlavní výhodu je považována efektivní analýza dat, kratší doba odezvy na generičtější dotazy a možnost předpřipravit typické pohledy (reporty). Nevýhody v porovnání s relačním přístupem jsou viděny nejčastěji ve vyšší ceně vytvoření databáze, složitějším rozhraní a údržbě databáze včetně komplikovanějšího plnění datových kostek a vyšších nároků na úložný prostor (Němec, 2014), (Golfarelli, 2009).

Hlavní rozdíl mezi multidimenzionalitou v prostředí OLAP a v prostředí relační databáze je spatřován v tom, že prostředí multidimenzionální OLAP je postaveno na ad-hoc logickém modelu, jenž je možno využít přímo k reprezentaci multidimenzionálních dat a operací. Naproti tomu v prostředí relační databáze se pro tyto operace musí použít množství spojujících operací (Golfarelli, 2009).

Kromě těchto dvou přístupů existuje kombinovaný hybridní přístup, ve kterém jsou uloženy velké objemy detailních dat v relačních databázích a agregovaná data, která jsou nejčastěji využívána v binárních OLAP kostkách (Pour, 2012). Výhodou řešení je, že jsou spojovány přednosti obou základních variant, a to především úroveň standardizace, možnost zvládnout velké objemy dat z relačního řešení a rychlost dotazů z multidimenzionálního řešení (Golfarelli, 2009).

2.4) Architektura a komponenty systému BI

V systému business intelligence se vyskytují řady jednotlivých komponentů, které jsou integrovány a využity v závislosti na potřebách konkrétní organizace. Integrací jednotlivých komponentů je vytvářena architektura systému BI, v nichž jsou obsaženy vazby mezi komponenty a jejich vzájemné návaznosti v rámci plnění cílů podnikového informačního systému (Němec, 2014).

Na architekturu můžeme nahlížet z různých úhlů pohledu například Novotný (2005) ji rozděluje na pět vrstev:

- *vrstva pro extrakci, transformaci, čištění a nahrávání dat* pokrývající oblast sběru a přenosu dat ze zdrojových systémů do datových struktur BI,
- *vrstva pro ukládání dat* zahrnující procesy ukládání, aktualizace a správy dat,
- *vrstva pro analýzy dat* je spojována s komponenty pro zpřístupnění a analýzu dat – reporting a dolování dat,
- *prezentační vrstva* - nástroje pro komunikaci koncových uživatelů s komponenty BI řešení,

- ve *vrstvě oborové znalosti* je obsaženo best-practices implementace BI řešení v konkrétní situaci.

Autoři Kimball a Ross (2013) pojímají architekturu do dvou druhů komponentů:

- procesy ukryté uživatelům jsou zahrnuty v *back room komponentech*, jako transformace, ukládání dat a podobně, v nich jsou obsaženy jsou tyto komponenty - datová tržiště, dočasné úložiště dat a zdrojové systémy,
- část systému, která je viděna koncovými uživateli, je představována *front room komponenty*.

Níže se blíže seznámíme s jednotlivými komponenty architektury systému BI:

- produkční a zdrojové systémy,
- dočasné úložiště dat,
- operativní úložiště dat,
- transformační nástroje,
- integrační nástroje,
- datová tržiště,
- centrální datové úložiště,
- OLAP,
- komponenty uživatelského rozhraní systému BI
- nástroje pro zajištění kvality dat,
- nástroje pro správu metadat,
- ostatní.

2.4.1) Produkční a zdrojové systémy

V nich jsou obsaženy záznamy obchodních transakcí při využití jejich struktury a formátu dat. Jejich hlavním úkolem je zajistit provoz podniku, jedná se o podnikové informační systémy (ERP), systémy řízení dodavatelského řetězce (SCM), systémy řízení vztahů se zákazníky (CRM) a další. Zpravidla je zobrazován jen jeden záznam v čase, typicky nacházíme jen málo historických údajů. V mnoha případech jsou poskytovány tyto systémy pro speciální účely jednotlivých aplikací a bez závazku jsou sdílena společná data, jako informace o produktech, zákaznících a další, s ostatními zdrojovými systémy organizace. Zdrojovými systémy jsou myšleny i systémy mimo společnost, například veřejně dostupné databáze, sociální média a další (Kimball a Ross, 2013),(Pour, 2012).

2.4.2) Dočasné úložiště dat (Data Staging Area - DSA)

Je používáno pro dočasné uložení extrahovaných dat z produkčních databází. Cílem tohoto uložení je zajistit přípravu a potřebnou datovou kvalitu před nahráním do datového skladu.

Data zde uložena lze charakterizovat jako detailní, neagregovaná, nekonzistentní a pouze aktuální (Pour, 2012).

2.4.3) Operativní úložiště dat (Operation Data Store - ODS)

Je využíváno pro integraci skupiny subjektivě orientovaných dat pocházejících ze zdrojových systémů do konzistentních dat určených pro operativní rozhodování. Data spravována tímto systémem jsou částečně konsolidována s malým nebo vůbec žádným časovým rozlišením. Je možno data modifikovat a následně synchronizovat se zdrojovými systémy nebo datovým skladem (Němec, 2014). Hlavní rozdíl mezi DSA a ODS je postaven na umožnění přístupu uživatelům, kdy do DSA uživatelé nemají přístup, naproti tomu ODS aktivně využívají (Tvrdíková, 2000).

2.4.4) Transformační nástroje (Extract, Transform, Load - ETL)

Jedná se o proces získávání dat jejich transformací a následného nahrání do datového skladu. Extrakce je dána jako první fáze v procesu získávání dat, při níž je využíváno čtení, porozumění zdrojovým datům, jež jsou extrahována do ETL systému pro jejich další zpracování. V druhé fázi procesu je uplatňováno množství transformací zahrnující čištění dat (opravování překlepů, ošetření prázdných hodnot, převedení formátů), spojování dat z několika datových zdrojů a odstranění duplicitních hodnot. Třetí poslední fází je strukturování (přiřazení umělých klíčů, popisů atd.) a nahrání dat do prezentační oblasti cílového dimenzionálního modelu (Kimball a Ross, 2013).

2.4.5) Integrační nástroje (Enterprise Application Integration - EAI)

Za integrační nástroje jsou považovány aplikace určené pro integraci primárních podnikových systémů a redukci počtu jejich vzájemných rozhraní. Jsou známy dvě místa jejich výskytu. *Na úrovni datové integrace*, kde jsou využívány pro integraci a distribuci dat, a *na úrovni aplikační integrace*, kde jsou upotřebeny především pro sdílení jistých vybraných funkcí aplikací. Rozdíl mezi nástroji EAI a ETL je zaznamenán v tom, že nástroje EAI jsou schopny pracovat v reálném čase. Jsou užívány především v transformační vrstvě, kde slouží k přenosu dat do datových úložišť v reálném čase (Pour, 2012).

2.4.6) Datová tržiště (Data Marty)

V datových tržištích shledáváme značnou podobnost s datovými sklady s tím rozdílem, že datová tržiště jsou určena pouze pro omezený okruh uživatelů (pro pobočku, oddělení).

Můžeme je chápat jako decentralizované datové sklady, které jsou problémově orientovány a jsou jimi umožňovány „ad-hoc“ analýzy dat (Novotný, 2005).

2.4.7) Centrální datové úložiště

Centrální datové úložiště se vyznačuje vlastnostmi „*integrované, subjektivě orientované, stálé a osahující časově rozlišený souhrn dat, uspořádaný pro podporu potřeb managementu*“ (Novotný, 2005, str. 32). Inmon (2005) podává bližší vysvětlení těchto pojmů:

- subjektivě orientované, data jsou rozdělována podle jejich typu nikoliv podle aplikací, kde byly vytvořeny,
- integrované, data jsou shromažďována z několika zdrojů a útvarů do jedné výsledné formy v rámci celého podniku,
- stálé, jsou konstruována převážně pro čtení, zpravidla se žádná nová data nevytvářejí ani neaktualizují,
- časově rozlišené, kdy data jsou nahrávána pravidelně po několik let.

Architektury datového skladu

V předchozích částech jsme definovali jednotlivé komponenty pro uložení dat. Nyní se zaměříme na architekturu úložiště, tedy jak jsou tyto komponenty koncipovány a propojeny.

Architektury jsou podle Golfarelli (2009) založeny na pěti základních vlastnostech:

- *oddělenost* - analytické a transakční zpracování je nutno ponechat oddělené to té míry, jak jen je možné,
- *škálovatelnost* – HW a SW architektura by měla být jednoduše rozšiřitelná, především s ohledem na zvýšení objemu dat a počtu uživatelských požadavků,
- *rozšiřitelnost* – architektura by měla být schopná pojmout nové aplikace a technologie bez nutnosti přestavby celého systému,
- *bezpečnost* – základem je sledovatelnost přístupu s ohledem, že se jedná o úložiště strategických dat,
- *spravovatelnost* – správa datového skladu by neměla být příliš složitá.

Architektura datového skladu se rozlišuje v podstatě na základě dvou hlavních proudů a jejich modifikací a vzájemných spojení. První z těchto proudů je založen na názoru Williama H. Inmona, který je představován centralizovanou koncepcí. Druhý názor Ralpha Kimballa je specifikován integrovaným decentralizovaným přístupem. V současné praxi je nejčastěji uplatňován hybridní přístup kombinující výhody obou výše uvedených (Němec, 2014).

a) Kimballova DW/BI architektura

Tuto architekturu můžeme v literatuře nalézt pod různými názvy, například jako architekturu nezávislých datových tržišť nebo decentralizované pojetí architektury.

Její přístup byl definován v 80. letech a je založen na principu, kdy jsou analytická data rozmístěna podle útvarů podniku bez důrazu na sdílení a integraci informací napříč společnostmi. V rámci architektury jsou postupně vytvářena jednotlivá datová tržiště, vždy pro specifický útvar podniku. Pokud je nastolena potřeba získat stejné informace, jsou zpravidla vytvářena téměř duplicitní řešení bez vzájemných propojení s již existujícími datovými tržišti.

Výhoda tohoto řešení je dána rychlostí realizace BI, kdy je často uplatněno přírůstkové budování a jednotlivá řešení jsou dodávána v relativně krátkých intervalech. Je docíleno efektivnějšího řízení a kontroly. V datovém tržišti jsou uchovávána jak atomická, tak i agregovaná data, jsou jím umožňovány jednodušší aplikace, tvorby dotazů, orientace uživatele apod.

Naopak nevýhody jsou spatřovány v obtížnějším zajištění celkové integrace řešení, výstupy v podobě jednotného reportingu jsou považovány za finančně i časově náročnější. Jak již bylo výše zmíněno, je zde možnost rizika vzniku obsahového překrývání a duplicita v jednotlivých komponentech, čímž je zvyšováno i riziko vyšších nákladů na provoz (Kimball a Ross, 2013), (Pour, 2012).

b) Modifikovaná verze Kimballovy architektury

Tato architektura je známa jako sběrníková architektura. Její vznik je připisován vývoji Kimballovy architektury v 90. letech zavedením řešení, v němž je obsaženo více datových tržišť a sdílených dimenzí, které byly sdíleny informacemi napříč jednotlivými tržišti.

Výhody modifikované verze - nízká doba vývoje a jeho nižší náročnost pohybující se v řádech 3-6 měsíců v závislosti na velikosti řešení, lepší kontrola nad rozpočtem a rychlejší návratnost investice, rozsah tržišť je většinou omezen z důvodu nižších nároků na HW, počet uživatelů se pohybuje pouze v desítkách z důvodu zaměření se pouze na jednu organizační jednotku.

Nevýhody modifikované verze - rostoucí náklady v případě duplikace datových pump, jsou navýšeny celkové náklady, pokud je tvořeno řešení napříč různými platformami mezi jednotlivými datovými tržišti, při sdílení dimenzí je zvýšeno riziko vzniku bezpečnostních incidentů (ochrana osobních údajů a osobního vlastnictví), uchovávání dlouhé historie může být problematické z důvodu nárůstu množství dat při procesně orientovaných multidimenzionálních strukturách (Němec, 2014).

c) Architektura Billa Inmona

V literatuře je uváděná i pod názvem centralizovaná architektura. Koncepce je založena na existenci jediného konsolidovaného úložiště (celopodnikového datového skladu). Realizuje se na principu extrakce dat z provozních zdrojových systémů a zpracováním pomocí ETL systému. Následně jsou nahrána atomická data do databáze, a to ve třetí normální formě. Je zde tedy vyžadována normalizace, čímž jsou narušovány integrace – tím je představován zásadní rozdíl oproti Kimballově architektuře (Kimball a Ross, 2013).

Výhody architektury řešení jsou známy svou flexibilitou, integrovaností a je možno ji využít pro náročné analytické úlohy, není docházeno k duplikaci komponentu BI (Pour, 2012), (Novotný, 2005). Je snadné provádění pohledů napříč organizací, jednotné řešení kvality a znovupoužitelnosti dat (Němec, 2014).

Nevýhody v řešení jsou představovány delší dobou trvání, finanční (až kolem 20 milionů – dle knihy Němec, 2014) a časovou náročností, dále menší možností sledovat návratnost investic, při změně zdrojových systémů nebo požadavků na systém, během implementace jsou považovány změny těchto realizací za náročné (Pour, 2012). Současně s většími nároky na databázovou platformu a menší škálovatelnost (Němec, 2014).

d) Modifikovaná verze Inmonovy architektury

Modifikovaná verze bývá často označována jako „hub and spoke“ nebo hybridní architektura. Je kombinována datovým skladem a datovým tržištěm napojeného na datový sklad, neboli přístupu Kimballa a Inmona. Podobnost je sledována s Kimballovou prezentační oblastí, ve které jsou umístěna data dimenzionální, atomická, založena na procesech a přizpůsobena podnikové sběrníkové architektuře (Kimball a Ross, 2013).

Výhody jsou tvořeny především větší flexibilitou a škálovatelností systému, možností vytvářet další datová tržiště (i v jiné než dimenzionální formě pro dolování dat), je minimalizována duplicita v klíčových komponentech (Pour, 2012) a je vytvářena jednoduchá tvorba konsolidovaných celopodnikových reportů (Němec, 2014).

Nevýhody vidíme podobné jako u nemodifikované verze. Je odstraňován problém menší škálovatelnosti. Je nezbytné pokrýt všechny uživatelské požadavky najednou, a s tím je spojena zdoluhavá analýza a dokumentace (Němec, 2014).

2.4.8) OLAP (On-line Analytical Processing)

Multidimenzionální koncept OLAP jsme si již definovali v kapitole 2.3, kde jsme se zabývali multidimenzionalitou a multidimenzionálním pohledem na data. Jednalo se o dvě základní varianty a jednu kombinovanou s oběma variantami. Konkrétně byl popisován:

- multidimenzionální OLAP (MOLAP),
- relační OLAP (ROLAP),
- hybridní OLAP (HOLAP).

2.4.9) Komponenty uživatelského rozhraní systému BI

Můžeme rozdělit na tři hlavní oblasti reporting, manažerské aplikace pro vrcholový a střední management a dolování dat.

Reporting je formulován jako funkcionalita umožňující vytvářet automatický interaktivní výstup, v některých případech menší automatizace i poloautomatické a manuální statistické výstupní sestavy. Jsou jím poskytovány informace nezbytné k efektivnímu řízení procesů v organizaci (Němec, 2014). Rozlišujeme dva druhy reportingu - *standardní*, spouští se předpřipravené dotazy, a *ad hoc reporting*, kdy aplikujeme jednorázově uživatelem vytvořené dotazy (Novotný, 2005).

Manažerské aplikace pro vrcholový a střední management - do této kategorie jsou přiřazovány systémy MIS, EIS a ES, které jsme definovali v kapitole 2.2. K systémům EIS jen v této části doplníme, že byly přesunuty vývojem od počítačového zpracování velkého množství dat k řešení běžných a pokročilejších rozhodovacích a informačních potřeb, především s ohledem na top management organizace. Je jimi umožňována tvorba strategických analýz, rozhodování a strategického plánování, využívající princip multidimenzionality k predikci a analýze trendů. Kromě toho byly započaty integrace s dalšími nástroji pro strategické řízení a od poloviny 90. let 20. století se začaly objevovat i v České republice a postupně byly vyvíjeny nové systémy již pod pojmem BI. V současnosti jsou EIS plně přeměněny do reportingové součásti BI v podobě dashboardů a dalších grafických výstupů, postavených na OLAP technologii (Němec, 2014).

Dolování dat je představována sada analytických metod a modelů sloužících pro analýzu dat obsažených v rozsáhlých databázích. Za jejich účel je považováno automatizované objevování původně neznámých vztahů mezi daty a skrytými vzorci, trendy, korelacemi a umožnit tak předpovídat budoucí chování v rámci zkoumané oblasti podnikové reality pomocí

rozpoznávání vzorů a pokročilé matematiky a statistiky (Vercellis, 2009). Dolování dat je založeno na matematických a statistických technikách, jako rozhodovací stromy, neuronové sítě, generické algoritmy, clustering a klasifikace apod. (Novotný, 2005).

2.4.10) Nástroje pro zajištění kvality dat

Data s nízkou kvalitou jsou někdy vyskytována ve zdrojových systémech (problémy v syntaxi, sémantice, chyby uživatelů). Současně s důsledkem integrace několika datových zdrojů do jednoho systému je možno dospět k výraznému narušení kvality dat. Chybná data musíme opravit a zajistit tak kvalitu rozhodování založeného na těchto datech.

V rámci projektování ETL procesu je předpokládána nutnost analýzy kvality vstupních dat, neboli k profilování dat, kdy se snažíme odhalit nekonzistence a defekty v datech. Následně jsou navrhovány rutiny pro čištění a transformaci dat do datového skladu (Němec, 2014). Tvrdíková (2008) definuje následující vlastnosti kvalitních dat:

- **dostupnost** – možnost přístupu uživatele k datům v okamžiku jeho potřeby, je zahrnována dostupností v čase, v místě, v požadované struktuře a v požadovaném formátu,
- **přesnost** - data a jejich přesnost jsou ověřována nejčastěji z externích zdrojů (jako příklad si uvedeme PSČ, zda je správně řazeno zadanému městu),
- **úplnost** – zabývá se kontrolou dispozic všech dostupných informací v rámci daného kontextu,
- **konzistence** - je jí kontrolován výskyt možných problémů v porušení standardů nebo vazeb mezi daty.

Někteří autoři, například Novotný (2005), k tomu ještě přidávají soulad, unikátnost a integritu.

Kontrola kvality dat může být provedena ve třech částech - *během provozu transakčních aplikací*, kdy uživatelé sami dbají na kvalitu dat, *uvnitř úložiště systému BI* s pomocí interních utilit pro opravy, *během návrhu systému BI*, kdy kontrola je prováděna v analýze současného stavu a v rámci profilování zdrojových dat (Němec, 2014).

2.4.11) Nástroje pro správu metadat

Metadata se často definují jako data o jiných datech. V oblasti datových skladů jsou pokládána za velmi důležitá a jsou využívána pro specifikaci zdrojů, hodnot, užití a vlastností DW. Je jimi popisováno, jak jsou data upravena a zpracována v každé vrstvě architektury. Mimo datový sklad jsou upotřebována k popisu významu a struktury dat ze zdrojových systémů.

Kelly rozděluje metadata do dvou kategorií. Kategorie jsou založeny na tom, jak systémový administrátoři a koncoví uživatelé prohledávají metadata. Rozděluje je tedy na interní a externí.

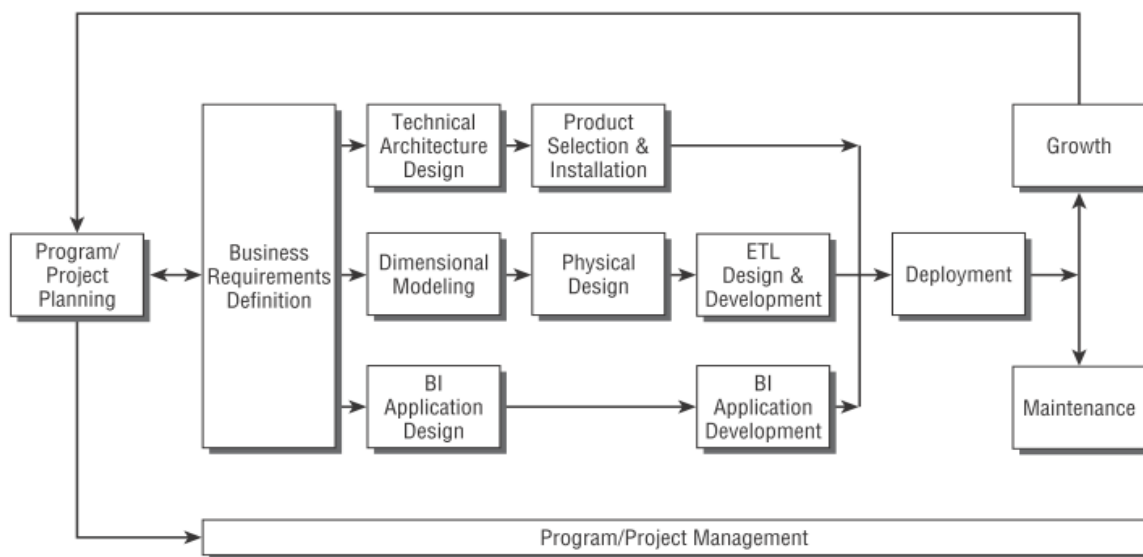
Systémovými administrátory jsou využívána **interní** metadata pro definování datových zdrojů, transformačních procesů, logického a fyzického schématu, omezení a uživatelských profilů.

Externí metadata jsou pokládána za relevantní pro koncové uživatele – definice, standardy dotazování, jednotky měření a agregace. Jsou uložena v metadatovém úložišti (Golfarelli, 2009).

Metadata by podle výše uvedeného vymezení měla poskytovat následující informace: o struktuře úložišť BI, genealogii dat, statistikách využívání úložišť a dokumentaci obecného významu úložišť z pohledu aplikační domény (Němec, 2014).

2.5) Metodika budování systému Business Intelligence

Na základě předchozích teoretických východisek v této podkapitole bude představen proces budování systému BI pomocí diagramu životního cyklu od autora Kimballa viz diagram životního cyklu číslo: 2-3. Zaměříme se především na střední (hlavní) linii tedy na analýzu požadavků, dimenzionální modelování, fyzický návrh, návrh ETL a vývoj.



Obr. 2-3: Kimball diagram životního cyklu
Zdroj: (Kimball a Ross, 2013, str. 404)

2.5.1) Analýza požadavků

Existuje několik přístupů k analýze uživatelských požadavků. Dané požadavky jsou představovány otázkami a jejich odpověď budou hledat uživatelé v rámci BI aplikací. V této části práce uvedeme některé způsoby získávání požadavků (Novotný, 2005).

Počáteční (kick-off) workshop, pro seznáení zainteresovaných osob s plánem analýzy, jejími cíli, časovým plánem a nároky na pracovníky podniku (Novotný, 2005).

Interview - Kimballův přístup využívá dvou základních forem a to rozhovoru nebo skupinového sezení. Rozhovory jsou dělány s jednotlivci nebo v malých homogenních skupinách, kdy za výsledek je pokládán velmi detailní seznam specifikací. Naproti tomu skupinové sezení je zahrnuto velkými heterogenními skupinami, které jsou vedeny zprostředkovatelem, jenž zajišťuje, aby byly strany navzájem pochopeny. Tento přístup podporuje kreativní brainstorming.

Cílově založená analýza požadavků je používána pro navrhování datových tržišť. Tento přístup je založen na dvou odlišných pohledech, a to na modelování rozhodnutí, které je zaměřeno na požadavky podnikového rozhodování, a organizační modelování, s orientací na stakeholdery (Golfarelli, 2009).

K dalším cestám k analýze požadavků jsou připisovány naslouchání a pozorování pracovníků při práci, dotazníkové šetření, studium dostupné dokumentace, procedury a podobně, agilně orientované metody a metoda uživatelských požadavků známější pod anglickou zkratkou „user story“ (Němec, 2014).

V rámci práce se předpokládá využití interview se zákazníkem a na jeho základě vytvoření seznamu požadavků, jejich následné schválení zákazníkem. V případě potřeby pak použijeme další metody.

2.5.2) Dimenzionální modelování

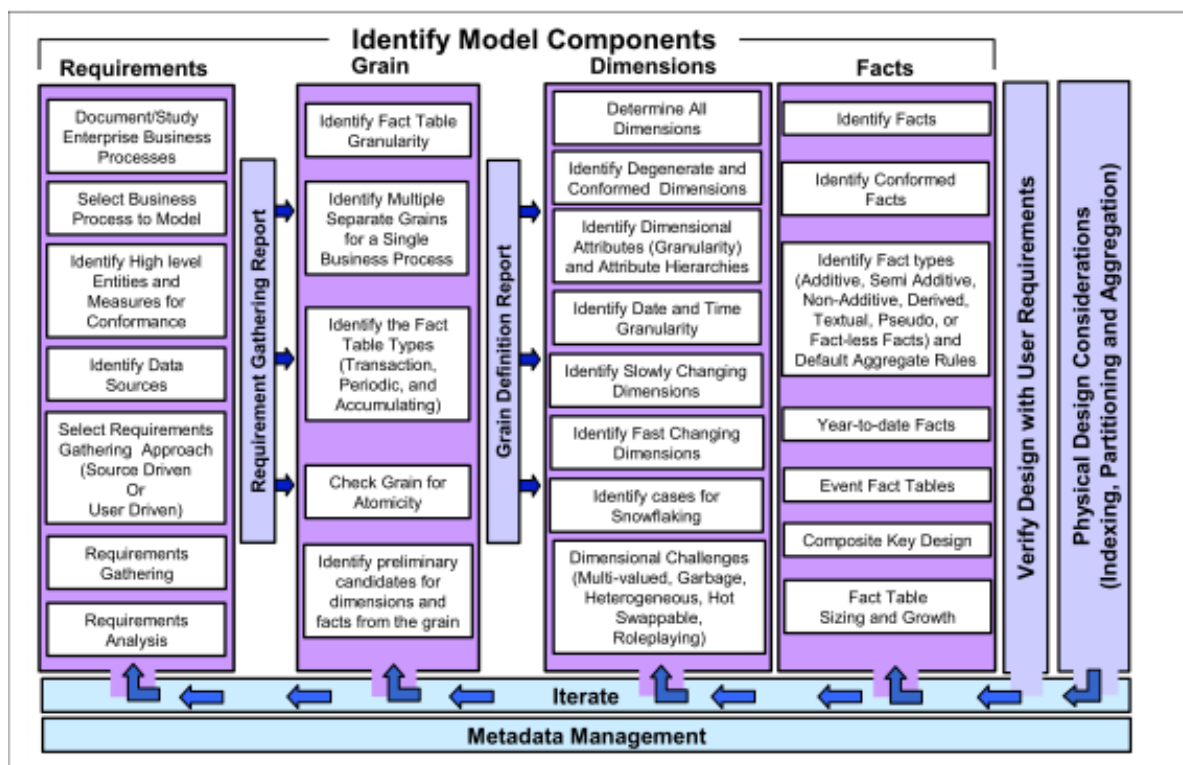
Je chápáno jako technika analýzy vedoucí k efektivnímu návrhu struktury a obsahu informačního modelu pro potřeby podpory rozhodování v organizaci. Za výsledek je brán návrh metrik a dimenzí včetně atributů. Výstupy této techniky je možno transformovat v podobě multidimenzionálního datového modelu. Podle Němce (2014) můžeme tuto analýzu označit za techniku sémantického modelování, z toho důvodu, že sémantická úroveň je určena k vymezení základní filozofie obsahu budoucího datového modelu, k čemuž je technika dimenzionální modelování z principu přímo určena (Němec, 2014).

Autor diplomové práce se ztotožňuje s názorem Němce (2014) ohledně sémantické úrovně modelování. Pokud by ale měl vycházet z rozdělení na tři techniky modelování (sémantickou, konceptuální a logickou), jak uvádí Kaluž a Kalužová (2012), přikláněl by se k technice sémanticko-konceptuálního modelování z důvodu, že výsledek dimenzionálního modelování můžeme shrnout v rámci konceptuálního modelu.

Dimenzionální modelování je složeno ze čtyř kroků podle Kimball a Ross (2013):

- výběr podnikového procesu** - uživatelé ho chtějí analyzovat a dlouhodobě vyhodnocovat pomocí systému BI,
- deklarace podrobnosti informací o podnikovém procesu** - především v jaké úrovni granularity chceme proces sledovat,
- identifikace dimenzí** - zde určujeme popisné informace k jednotlivým faktům, seznamy popisných atributů včetně definice hierarchie atributů,
- identifikace a výběr numerických faktů (metrik, ukazatelů)** - stanovení základních informací ve smyslu „co měříme a jak to měříme“ u jednotlivých procesů s cílem nalezení metrik a s přihlédnutím na určenou úroveň granularity.

Na základě znalosti Kimballova procesu budeme dále postupovat dle detailního popisu jednotlivých sub kroků dle Ballard a kol. (2006). Obrázkem 2-4 jsou znázorňovány následující jednotlivé kroky.



Obr. 2-4 Životní cyklus návrhu dimenzionálního modelu
Zdroj: (Ballard, 2006, str. 104)

a) Identifikace požadavků podnikových procesů

Je zahrnována výběrem podnikových procesů pro dimenzionální model založeného na výběru a požadavcích na podnikové procesy. Výběrem jednoho podnikového procesu je vyžadováno upřednostnění na základě kritérií, jako významnost podnikového procesu, kvalita

dat zdrojového systému, realizovatelnost a komplexnost. Uvedené kroky nyní blíže rozdělíme do několika fází.

Vytvoření celopodnikového seznamu procesů

V této první fázi vytvoříme kompletní seznam podnikových procesů včetně identifikace základních informací, například komplexnost zdrojového systému, dostupnost dat, kvalita dat a strategický význam. Pro určení podnikových procesů můžeme využít například techniku pravidla 5W-1H, kterým jsou poskytovány odpovědi na otázky – kdy, kde, kdo, co, jak a proč.

Identifikace podnikových procesů

Během fáze identifikujeme podnikový proces, který budeme modelovat pomocí dimenzionálního modelu, a to na základě ohodnocení výše uvedených kritérií jednotlivých podnikových procesů.

Identifikace entit a metrik

V tomto kroku vymezíme hlavní entity zahrnuté v každém procesu. S cílem nalézt takové, u kterých jsou objevovány společné prvky pro několik podnikových procesů. Takto vybrané entity využijeme napříč všemi dimenzionálními modely. Pro sjednocení pohledů na dané entity je považováno za nezbytné odsouhlasit definici sdílených entit jednotlivými podnikovými útvary.

Identifikace zdrojů dat

V této fázi se zabýváme identifikací zdrojů dat spojených s podnikovými procesy včetně určení vlastníka zdroje, cesty ke zdrojovému systému a platformy.

Výběr přístupu ke sběru požadavků

Přesně vymezit podnikové požadavky je pokládáno za velmi obtížné, požadavky organizace se zpravidla v čase mění. I přes omezení definujeme současné požadavky pro konstrukci dimenzionálního modelu (Ballard, 2006).

Abychom zjistili, které požadavky a informace jsou dány jako dostatečné pro modelování, odpovídáme si položenými otázkami (Ballard, 2006):

- kdo/co jsou zájmové subjekty (lidé, skupiny a organizace),
- které funkce mají být analyzovány,
- proč jsou vyžadována data,
- kdy je pokládáno za nutné data zaznamenávat,

- kde geograficky a organizačně,
- jak měříme výkonnost analyzovaných funkcí,
- jak je měřena výkonnost podnikových procesů, kterými faktory je ovlivňován úspěch nebo neúspěch,
- jaká je známa metoda distribuce informací,
- jaký typ informací je postrádán pro analýzy a rozhodování,
- jaké kroky jsou nyní prováděny pro odstranění chybějících informací,
- jaká bude požadována úroveň detailu pro analýzu dat.

Rozlišujeme dvě metody získávání požadavků, a to řízení zdroji dat a řízení uživateli.

Řízení zdroji dat - metoda založena na definování požadavků pomocí zdrojových dat z produkčních systémů. K tomu se využívá analýza entity-relationship modelu (E/R) zdrojových dat nebo dat samotných. Hlavní výhoda uvedeného přístupu je viděna v tom, že od začátku je disponováno se všemi daty. Druhá výhoda je dána v minimalizaci potřebného času uživatelů na začátku projektu.

Naopak nevýhody přístupu jsou vyplývající z minimálního zapojení uživatelů, což je směřováno ke zvýšení rizika zahrnutí nesprávných požadavků. V závislosti na velikosti zdrojových dat a dostupnosti E/R modelů by mohl být tento přístup časově náročným. Za další nevýhodu se pokládá nevyužití současných nedostupných dat, která mohou být mnohdy potřebná pro splnění hlavních uživatelských požadavků, převážně se jedná o externí data.

Ballard (2006) uvádí dva typy použití tohoto přístupu. První je využíván při vytváření poměrně obsáhlého seznamu hlavních dimenzí a druhým jsou objevovány nové oblasti datového skladu při analýze vztahů zdrojových dat.

Řízení uživateli - metoda je založena na definování požadavků pomocí zjišťování požadovaných funkcí od uživatelů. To je většinou provedeno na základě série rozhovorů a porad s uživateli. Přednost zmiňovaného přístupu je nalézána v jeho soustředěnosti se na to, co je opravdu potřebováno, a není směřován jen na to dostupné. Obecně je přístup vyznačován menším rozsahem než předchozí metoda řízení zdroji a jsou jím poskytovány datové sklady nebo datová tržiště v kratším čase.

Negativa pak spočívají v tom, že očekávání uživatelů musí být pozorně řízena a korigována. Uživatelé musí jasně rozumět, že některá data, jež jsou jimi vyžadována, nemusí být dostupná z mnoha důvodů. Další nevýhoda je spojována s úzkým zaměřením uživatelů, kdy nejsou vždy

zahrnuta vhodná data, které jsou obsaženy v produkčních systémech (Ballard, 1998), (Golfarelli, 2009).

Získávání požadavků

Po shromáždění požadavků a jejich dokumentaci se zaměřujeme na studium samotného procesu a analýzu informačních aktivit, které uživatelé vykonávají. Je bezpodmínečně vyžadováno, aby bylo získávání požadavků soustředěováno na dvě hlavní věci – co bylo analyzováno a jaká jsou vytyčena hodnotící kritéria.

Získávání požadavků se orientuje na pochopení modelované problémové oblasti. Současně je nezbytným cílem určit způsob, jakým bude pracováno s historií, tedy jakým způsobem budeme nahrávat změněná data.

Analýza požadavků

V rámci analýzy jsou hlouběji prozkoumávány metriky a entity získané v předcházejících fázích. Kromě toho zjišťujeme přidružené hierarchie entit. Za výstup těchto kroků je rozuměna zpráva o požadavcích obsahující (Ballard, 2006):

- seznam podnikových procesů,
- nastavení priorit podnikovým procesům,
- společné entity a metriky napříč podnikovými procesy,
- identifikaci modelovaného podnikového procesu,
- seznam datových zdrojů,
- seznam business požadavků,
- analýzu požadavků,
- identifikované entity a metriky.

b) Deklarace podrobnosti informací o podnikovém procesu (granularity)

V tomto druhém kroku je specifikováno, čím jsou přesně představovány jednotlivé řádky tabulky faktů, tudíž jaká je dána úroveň detailu (granularity) faktové tabulky a celého dimenzionálního modelu. Nejvhodněji je vyjadřována granularita pomocí obchodních podmínek, například: „Jeden řádek vyjadřuje bankovní účty za každý měsíc“ (Kimball a Ross, 2013).

Granularita faktové tabulky

Informace pro určení granularity faktové tabulky lze vyvodit například z dokumentů zpracovaných při získávání podnikových požadavků. Zvolení správné úrovně granularity je stanovováno stupněm detailu informací, které jsou obsaženy v rámci dimenzionálního modelu. Tento krok je pokládán za nejdůležitější v rámci celého návrhu dimenzionálního modelu. Kromě stanovení úrovně detailu všech datových dimenzí je nezbytné zvolit optimální úroveň detailu datumové dimenze, kdy se rozhodujeme mezi uložením ve dnech, týdnech, měsících nebo letech.

Identifikace vícenásobných oddělených granularit pro jeden podnikový proces

Fakta jsou hodnocena jako jedno z nejpodstatnějších kritérií při volbě jednoduché nebo vícenásobné tabulky faktů. Abychom si nejvhodněji vybrali z těchto možností, musíme porozumět dimenzionalitě faktů. Setkáváme se třemi nejčastějšími příčinami pro využití více tabulek faktů.

Pokud jsou fakta vedena za jednotlivé položky a současně i za agregované. Druhou variantou je vyznačována přítomnost několika zdrojových systémů, kdy každý systém je navrhován pro specifický účel. Za třetí možnost považujeme přítomnost nesouvisejících podnikových procesů.

Vícenásobná granularita faktové tabulky je řešena i možností, kdy je přidáván sloupec zvaný „vlajka granularity“. Tímto sloupcem jsou identifikována data nebo řádky ve faktové tabulce na denní, měsíční, kvartální nebo roční bázi. I přes možnost užití výše zmíněného přístupu jej autor nedoporučuje a raději navrhuje volit více faktových tabulek a schémat hvězdy (Ballard, 2006).

Identifikace typů faktové tabulky

V této části si určíme typ faktové tabulky, jež je zahrnuta do návrhu dimenzionálního modelu. Faktové tabulky můžeme rozdělit do tří skupin na transakční, periodické snímky a sumarizační snímky (Němec, 2014), (Golfarelli, 2009).

- *Transakční* - nejdetailnější typ tabulky faktů. Záznamy jsou stavěny na stejnou úroveň detailu jako v produkčních systémech. Frekvence záznamů odpovídá provádění jednotlivých transakcí.
- Údaje za určité pevně dané období (měsíc, kvartál, rok) jsou sumarizovány *periodickými snímky*, data jsou ukládána do BI systému v předem určených časových intervalech.

- *Sumarizační snímky* jsou užívány pro zjištění stavu za určité období, které nemusí být pevně stanoveno. Každý řádek tabulky je spojen s určitým výskytem aktivity v procesu. Jako jedinými jsou jimi umožňovány i průběžné aktualizace záznamů v závislosti na změnách stavů procesů.

Kromě těchto uvedených třech skupin existuje jeden speciální případ, a to bezfaktová tabulka faktů. Bezfaktovou tabulkou faktů je myšlena faktová tabulka obsahující pouze cizí klíče a žádná fakta (Ballard, 2006).

Ověření atomičnosti stanovené granularity

V tomto kroku přezkoumáme atomičnost úrovně detailu, abychom zjistili, zda jsou ukládána data na co nejpodrobnější úrovni. Vycházíme z toho, že čím detailněji jsou konkretizovány dimenze, tím detailnější jsou jimi poskytovány informace. Ovšem jsou nacházeny jisté kompromisy při zvažování granularity, například výkonnost versus množství dat, schopnost přístupu k datům na detailní úrovni versus výkonnost a cena za uložení dat.

Zvolením vhodné úrovně detailu dat jsou významně ovlivňovány velikosti dat v datovém skladu a současně je určována schopnost datového skladu uspokojovat dotazy uživatelů. Schopnost odpovědět na různé typy dotazů se zmenšuje díky nízké granularitě. Naopak prostřednictvím velmi vysoké granularity jsou prováděny jakékoliv dotazy za cenu zvýšení nároků na úložný prostor a výkonnost. Při rozhodování o úrovni detailu musíme pokaždé zvažovat kompromis mezi náklady na objem dat a schopnost zodpovědět dotazy.

Identifikace předběžných kandidátů pro dimenze a fakta vycházejících z granularity

Definujeme předběžné dimenze a tabulky faktů pouze z definice granularity, neprovádíme tedy žádnou detailní analýzu. Jakmile jednou vhodně definujeme granularitu, jednoduše můžeme nalézt předběžné dimenze a fakta, které si rozšíříme v části věnované identifikaci faktů a dimenzí (Ballard, 2006).

c) Identifikace dimenzí

V následujících osmi krocích identifikujeme jednotlivé dimenze a jejich prvky včetně určení časové granularity a způsobu řešení změn v dimenzích.

Určení všech dimenzí

Identifikovány jsou dimenze odpovídající definované granularitě. V dimenzích jsou obsaženy atributy a v některých z nich jsou obsaženy popisné informace. Jiné jsou využívány

ke specifikaci sumerizace faktových tabulek a jsou jimi poskytovány užitečné informace (Ballard, 2006).

Identifikace degenerovaných a sdílených dimenzí

V ***degenerovaných dimenzích*** nejsou obsaženy žádné atributy. Jejich výskyt je omezován pouze na základě příslušného atributu v tabulce faktů při neexistenci dimenzionální tabulky. Nejsou pokládány za typické dimenze, jsou často reprezentovány jednoduchými transakčními čísly umístěnými v tabulce faktů. Jsou využívány například u časové dimenze ve tvaru 20120610, kterou je jednoznačně vyjadřován rok – měsíc – den nebo číslo faktury (Pour, 2012).

Sdílené dimenze mohou být používány k různým faktovým tabulkám namísto opětovného vytváření stejných dimenzí. Dvě dimenze jsou sdíleny, pokud je sdílen jeden nebo více atributů, které jsou vyvozeny ze stejné domény. Tak můžeme kombinovat informace v jednom reportu napříč několika oddělenými faktovými tabulkami (Kimball a Ross, 2013), (Ballard, 2006).

Identifikace dimensionálních atributů, granularity a hierarchií

V této fázi naplníme vytvořené dimenze jednotlivými atributy. Než tak učiníme, je pokládáno za nezbytné určit granularitu jednotlivých dimenzí pro zachování stejné úrovně podrobnosti jednotlivých atributů v dimenzích. K dalším potřebným úpravám je přiřazováno vložení speciálního řádku „neznámí“ pro vyhnutí se prázdným (null) hodnotám ve faktové tabulce. Vložením prázdných hodnot do faktové tabulky by došlo k porušení referenční integrity a mohlo by dojít ke spojení na prázdných hodnotách klíčů.

Hierarchie je představována sérií 1:n vztahů. Skládá se obvykle z různých úrovní, kdy každá odpovídá atributu dimenze (Ballard, 2006). Kimball a Ross (2013) rozlišují čtyři typy hierarchie, které jsou uváděny níže.

Fixovaná hloubka hierarchie je prezentována sérií vztahů m:n, jako příklad můžeme uvést produkt ve vztahu ke značce, kategorii a k oddělení. Když je definována fixní hloubka hierarchie, její úroveň by se měla projevit jako oddělené pozice atributů v tabulce dimenzí. Pokud hierarchií není splňována podmínka série vztahů 1:n nebo počet úrovní proměnných, je vhodnější využít členitou hierarchii.

Mírně členitá / hierarchie hloubkově proměnlivá - u tohoto typu hierarchie nemáme pevně určené množství úrovní, ale rozsah hloubky je vnímán jako malý. Například geografická hierarchie je vytyčována rozsahem hloubky od tří do šesti úrovní.

Členitá / hierarchie hloubkově proměnlivá s hierarchií vazebních tabulek - hierarchie neurčené hloubky, jsou obtížně modelovány. Řešení je přinášeno zavedením tzv. vazební tabulky, ve které jsou obsaženy řádky pro každou možnou cestu v členité hierarchii a je ní umožňován přesun všech úrovní hierarchie pro zpracování standardním SQL.

Členitá / hierarchie hloubkově proměnlivá s atributy cesty - využitím vazební tabulky pro proměnlivě členitou hloubku hierarchie se můžeme vyhnout pomocí atributu cesty ve formě řetězce v tabulkách dimenzí. Atributem cesty je vyplňován každý řádek dimenze speciálně kódovaným textovým řetězcem s popisem kompletní cesty s nejvyšším významem v hierarchii včetně popisu konkrétního řádku dimenze.

V dimenzionálních tabulkách může být obsaženo více hierarchií a atributů nebo sloupců patřící jedné, více nebo žádné hierarchii (Ballard, 2006).

Identifikace granularity času a data

Identifikace těchto dvou veličin je pokládána za velmi důležitou, protože je ní pomáháno určit granularitu celého dimenzionálního modelu a informací v něm uložených. Pro snížení rozsahu dimenzionální tabulky je vhodněji oddělován čas od data, kdy každý den by byl zaznamenáván v datumové tabulce $60 \times 24 = 1440$ řádky navíc (Pour, 2012).

Dimenzionální model je zahrnován několika datumovými dimenzemi, které obvykle využívají konceptu zvaného „dimenze více rolí“ implementovaného pomocí cizích klíčů v tabulce faktů. Tento koncept se uplatňuje, pokud potřebujeme zaznamenat například datum objednání a datum obdržení, kdy oba údaje jsou charakterizovány stejnou granularitou. Místo využití dvou oddělených datumových tabulek se vytváří dva pohledy nad jednou datumovou tabulkou (Pour, 2012), (Ballard, 2006).

Identifikace pomalu se měnících dimenzí (Slowly changing dimension – SCD)

V nynější fázi identifikujeme pomalu se měnící dimenze, které jsou používány pro zaznamenání změn v dimenzích s ohledem na změny podmínek provozu organizace. Jedná se o změny typu přidání, rušení a přeměna názvů prvků či změny struktury. Němec (2014) rozlišuje tři základní a dva hybridní typy aktualizace dimenzí, viz níže.

SCD typ 1 (přepsání) - metodou není zachovávána žádná historizace, novou hodnotou jsou pouze přepisovány současné hodnoty. Po každé změně je nutné vygenerovat novou tabulku faktů, a tím jsou přepočítávány ukazatele podle nové metodiky.

SCD typ 2 (přidání nového řádku) - jsou vytvářeny nové řádky v dimenzionální tabulce s novým identifikátorem (ID, PK), kdy se vytváří duplicitní záznam, který se odlišuje od původního pouze jiným ID a novou hodnotou atributu. Výhoda přístupu je spatřována v zachování plné historizace a v odstraňování problému s potřebou vygenerování nové tabulky faktů. Častějšími změnami je docházeno ke vzniku redundance, a tím jsou vytvářeny větší nároky na úložný prostor a odezvu celého řešení.

SCD typ 3 (přidání nového atributu) - aktualizace u tohoto přístupu je řešena přidáním dalšího atributu do tabulky dimenzí. Přidaný atribut je používán pro uchovávání původní hodnoty atributu v rámci jednoho záznamu. Přednost tohoto řešení je viděna v zachycování změn bez zvýšení redundance dat. Naopak nevýhoda je založena na možnosti evidovat jen určitý počet změn. A proto nemůžeme řešení použít v případě častých změn atributů.

SCD hybridní typ 1 (SCD typ 4) - je specifikován tím, že je konstruován pro předem určený počet změn. Změny jsou evidovány pomocí oddělených atributů pro jednotlivé verze. V rámci této koncepce se předpokládá, že historické údaje budou postupně nahrazovány novějšími slovy a že bude udržována historizace jen několika posledních změn.

SCD hybridní typ 2 (SCD typ 6) - využívá se, pokud nejsme schopni dopředu definovat počet změn. V případě změn je jím přidáván nový záznam do tabulky dimenzí a pomocí atributu jsou jím zahrnovány současné a předchozí verze hodnoty.

Kromě výše uvedených jsou známy i další méně častější přístupy historizace údajů, založené například na vytvoření tabulky obsahující historii změn v dimenzionálních tabulkách nebo na přidání atributu časové platnosti (Němec, 2014).

Identifikace rychle se měnících dimenzí (Rapidly changing dimension - RCD)

Změny, které nemohou být zvládnuty předchozími přístupy. Pro tento typ dimenzí se taktéž užívá název rapidně se měnící dimenze, vyznačující se velmi častými změnami, zpravidla denními, týdenními nebo měsíčními.

S rychlými změnami dimenzí se nejlépe vypořádáme oddělováním rychle se měnících atributů do jedné nebo více samostatných dimenzí nazvaných **mini dimenze**. Ve faktové tabulce bude následně obsažen další cizí klíč pro mini dimenzi (Ballard, 2006).

Kimball a Ross (2013) k tomu navíc přidávají přístup nazvaný **SCD hybridní typ 3 (SCD 7)**, který se podobá typu 6 s rozdílem, že je odkázán na dva cizí klíče ve faktové tabulce. Je jím

přidáván řádek dimenze typu 2 s novou hodnotou atributu a zároveň je vytvářen pohled na aktuální řádky a hodnoty atributů (Kimball a Ross, 2013).

Identifikace případů sněhové vločky

V relačních databázích se často vyskytuje schéma sněhové vločky z důvodu normalizace. Pro dimenzionální modelování není dané schéma doporučováno z důvodu vlivu na pochopitelnost modelu a zvýšení nároků na výkonnost kvůli většímu množství spojovaných tabulek.

V jistých případech je ovšem využíváno i schématu sněhové vločky. Jedná se především o následující dva (Ballard, 2006):

- pokud je dimenzionální tabulka obsahuje dva nebo více atributů rozdílné granularity
- skupina atributů stejné dimenzionální tabulky je brána z rozdílných zdrojových systémů.

Identifikace speciálních případů dimenzí

V průběhu tvorby multidimenzionálního datového modelu občas nastávají situace, ve kterých, je potřeba zachytit specifické jevy svázané s požadavky na multidimenzionální pohled na data (Němec, 2014). V této části tyto nejčastější situace přiblížíme včetně jejich možného řešení. Ballard (2006) uvádí následující dva typy speciálních případů dimenzí:

Dimenze s více rolemi - je vyskytována v případě, kdy na jednu dimenzi je odkázáno množstvím cizích klíčů z tabulky faktů, kdy každý z nich je představován jinou rolí, například datum začátku a datum konce platnosti (Němec, 2014). Ballard (2006) předkládá možné řešení prostřednictvím dimenze, která je vyjádřena rozdílně ve faktové tabulce pomocí pohledů. Druhé řešení je spatřováno v užití více cizích klíčů, kterými je ukazováno na primární klíč v dimenzi.

Heterogenní dimenze - vyskytuje se u dimenzionálního modelování společností prodávající heterogenní produkt s odlišnými atributy stejným zákazníkům. Heterogenní produkty se vyznačují oddělenými jedinečnými atributy a není proto možno vytvořit jednu produktovou tabulku, která by byla schopna zvládnout uchovat tyto heterogenní produkty. Řešením je spojit všechny atributy do jedné velké dimenzionální tabulky nebo vytvoření oddělených dimenzí a faktových tabulek pro rozdílné heterogenní produkty (Ballard, 2006).

Následující čtyři speciální případy definují Kimball a Ross (2013), viz níže:

Okrajové dimenze (garbage dimenze) - transakčními obchodními procesy jsou standardně produkovány množstvím sloupců s nízkou kardinalitou, jako kódy, indikátory, statusy a vlajky. Než vytvářet oddělené dimenze pro každý takový atribut je považováno za vhodnější vytvořit jednu garbage/junk dimenzi, obsahující všechny sloupce pohromadě

Dimenze vyměnitelné za provozu - používáme, pokud je alternativně propojena jedna faktová tabulka s různými kopiemi stejné dimenze, které mohou být vyměněny při provádění dotazu. Každá verze vyměnitelné dimenze může mít rozdílnou strukturu a každá alternativní verze přistupuje do stejné faktové tabulky ale generuje rozdílný výstup. Pro implementaci musíme vypnout omezení referenční integrity mezi faktovou tabulkou a jednotlivými verzemi dimenze.

Vícehodnotové dimenze - standardně v každém dimenzionálním atributu by měla být obsažena jen jedna hodnota. Nicméně je známa existence situace, kdy tabulkou faktů jsou podporovány dimenze, které mají více atributů na nejnižší úrovni granularity faktové tabulky. Jako příklad si uvedeme pacienta, který má více paralelních diagnóz. Pokud se nemění granularita faktové tabulky, musí být propojena vícehodnotová dimenze s tabulkou faktů přes asociativní entitu zvanou „mostní tabulka“, jejímž obsahem je jeden řádek pro každou diagnózu (Kimball, 2004), (Kimball a Ross, 2013).

Auditní dimenze - je formulována jako speciální dimenze sestavená v pozadí ETL systému pro každou faktovou tabulku. Obsahuje meta data o události vytvoření konkrétního řádku faktové tabulky. V případě vzniku chyb je vytvářen řádek udávající podmínky vzniku chyby (Kimball a Ross, 2013), (Němec, 2014).

Němec (2014) uvádí další příklady, viz níže:

Dimenze s vazbou nadřazenosti a podřazenosti v attributech - jedná se o obdobu unárních relací, kdy například jeden uživatel je na úrovni juniora a druhý na úrovni team lídra pro více uživatelů. Oba uživatelé jsou následně uloženi v jedné tabulce.

Degenerovaná dimenze - její dimenze je zaznamenána pouze ve formě konkrétního atributu v tabulce faktů, jak jsme již definovali v kapitole věnované identifikaci degenerovaných a sdílených dimenzí.

Stavové dimenze - jsou obsahovány pouze popisy stavů, například stavy workflow, transakcí, objednávek a podobně.

Mini dimenze - vznik je dán rozdělením původní dimenze na dvě části, kdy v oddělené části se předpokládá velká četnost změn atributů (Němec, 2014).

d) Identifikace a výběr numerických faktů (metrik, ukazatelů)

Předběžná fakta jsme již určili z definice granularity, nyní budeme interaktivně a více detailně identifikovat fakta, která jsou pokládána za vhodná pro dané úrovně granularity.

Identifikace faktů

Zde identifikujeme všechna fakta odpovídající dané granularitě. Proces se skládá ze dvou částí:

- identifikace předběžných faktů pomocí úrovně granularity (již definováno v předchozích částech),
- identifikace faktů na základě detailní analýzy zdrojového E/R modelu.

Detailní analýza E/R modelu pro převod na model dimenzionální se skládá z následujících čtyř kroků:

- identifikace podnikového procesu v E/R modelu,
- identifikace vztahů m:n tabulek a jejich převedení na faktové tabulky,
- denormalizace zbývajících tabulek do plochých (flat¹) dimenzionálních tabulek,
- identifikace data a času z E/R modelu.

Identifikace sdílených faktů

Sdílená fakta jsou navržena pro stejné použití napříč několika datovými sklady nebo tržišti. Jakmile fakt identifikujeme, musíme ověřit, zda tento fakt již není evidován v datovém skladu nebo v jiném datovém tržišti, pokud je evidován, měli bychom využít již existující fakt (Ballard, 2006).

Identifikace typů faktů a agregačních pravidel

Ve faktové tabulce nacházíme několik rozdílných typů faktů, jednotlivé druhy si v této části přiblížíme. Němec (2014) rozeznává následující tři základní druhy faktů:

- ***aditivní fakta*** – ukazatele, kterými je umožňováno získat při agregacích korektní a smysluplné výsledky pro všechny zvolené dimenze, například prodej v kusech, náklady v Kč,

¹ **Flat dimenze** - - je pokládána za existující, pokud jsou vyznačovány hodnoty dimenze stejnou úrovní. Žádná hodnota není dítě nebo rodič jiné hodnoty (Oracle, 2001).

- ***semi-aditivní fakta*** – v případě agregace jsou údaje použitelné pouze pro atributy určité množiny dimenzí, jako příklad můžeme užít zásobu v kusech, kdy agregace je pokládána za smysluplnou pro atributy v dimenzi, ale produkty nejsou získávány žádné smysluplné a využitelné informace pro časovou dimenzi agregací,
- ***neaditivní fakta*** – agregací ukazatelů není dáván smysl v žádné situaci, výsledná informace by byla vnímána v případě agregace vždy jako nesmyslná, nejčastěji se jedná o agregace ukazatelů v procentech.

Ballard (2006), pak tento seznam rozšiřuje o další dva druhy fakt.

- ***odvozená fakta*** - jsou vytvořena vykonáváním matematických operací na několika jiných faktech, občas se označují jako vypočtená fakta a můžeme je uložit ve faktové tabulce,
- ***textová fakta*** - jsou složena z jednoho nebo více znaků, neměli bychom je uchovávat ve faktové tabulce, místo toho by měla být uložena v dimenzích.

Numerická fakta typu rok k datu (YTD, neboli year-to-date)

Numerické hodnoty obsahující agregace od začátku roku do aktuálního data. Pokud jsou zahrnuty agregované údaje do faktové tabulky, je zlepšována výkonnost a snižována komplikovanost při vytváření agregačních dotazů na sumarizovaná data (Ballard, 2006). Naopak Kimball a Ross (2013) považují za vhodnější spočítat tyto sumarizace až na úrovni BI aplikací nebo OLAP kostek, než aby byly ukládány ve faktové tabulce. Implementace je tedy postavena na OLAP aplikaci nebo funkcích jazyka SQL v pohledech a uložených procedurách.

Událostní faktové tabulky

Jsou využívány pro zaznamenávání událostí, jako počet kliků na webové stránce, docházka zaměstnanců. Událost jako pohyb uživatele na stránce nevede vždy ke vzniku faktu. Pokud chceme zaznamenávat takové události, kde se nevytváří fakta, používáme událostní faktovou tabulku obsahující neaditivní fakta nebo bezfaktovou faktovou tabulku.

Návrh složeného klíče

Primární klíč faktové tabulky je obvykle složen z několika cizích klíčů jednotlivých dimenzionálních tabulek, kdy každému cizímu klíči je příslušná jedna dimenzionální tabulka. Nicméně není bráno za povinné mít zahrnuty všechny cizí klíče ve složeném primárním klíči faktové tabulky. V některých případech ale ani kombinací všech cizích klíčů není garantována

jedinečnost a tehdy se využívá degenerované dimenze jako části primárního klíče faktové tabulky, kterým již je zajišťována jedinečnost (Ballard, 2006).

Odhad velikosti a růst faktové tabulky

Existuje množství přístupů pro odhad růstu faktové tabulky, níže jsou uvedeny dva z nich.

- ***Porozumění businessu*** - odhad je založen na předpokládaném tempu zaznamenávání jednotlivých záznamů, například u prodejní společnosti na velikosti prodejů a průměrné ceně zboží (Ballard, 2006).
- ***Na technickém pohledu*** - odhad je počítán velikostí růstu faktové tabulky pomocí výpočtu velikosti cizích klíčů, degenerovaných dimenzí, faktů a ročních přírůstků dat. Jakmile je spočítána celková velikost jednoho řádku faktové tabulky, vynásobíme ji očekávaným počtem řádků (příchozích dat) v rámci dané faktové tabulky (Rainardy, 2008).

Tímto krokem byl ukončen proces dimenzionálního modelování, výsledky by měly být zachyceny v rámci logického datového modelu v CASE nástroji. Konceptuální podoba struktury modelu by měla být v stávající fázi konzultována se zadavatelem práce pro posouzení splnění představ a požadavků, jinými slovy neboli provedena verifikace modelu (Ballard, 2006).

2.5.3) Fyzický návrh datového modelu

Po předchozí fázi návrhu dimenzionálního modelu se stává potřebou zmíněný model převést na model fyzický. Fyzický model je odlišován od logického modelu v detailech specifikovaných pro fyzické úložiště, jako jména sloupců, datových typů, deklarací klíčů a povolení nulových hodnot. V rámci současné podkapitoly jsou uváděny i operace zvýšení výkonnosti databází, zejména pomocí agregací, indexování a rozdělování tabulek (partitioning).

Agregační strategie

Agregace je známa jako proces počítání souhrnů dat z detailních záznamů faktové tabulky (Ballard, 2006). Každý datový sklad by měl obsahovat předkalkulované a předem uložené agregované numerické hodnoty. Snažíme se vyhnout rozdílné granularitě v jednotlivých faktových tabulkách a každá odlišná granularita agregace by měla být opatřena vlastní fyzickou faktovou tabulkou (Kimball a Ross, 2002).

Agregační navigace je vymežována jako software, kterým jsou zachycovány SQL požadavky a ty jsou softwarem transformovány pro užití nejlepších dostupných agregací.

Dostupnost agregační navigace je zařazována mezi další body pro zvážení při celkové agregační strategii. Funkcionalita je kladena mezi požadavky klienta a relačního databázového systému, je ní zahrnováno přesměrování SQL dotazů k vhodným předpřipraveným agregacím.

Indexační strategie

Indexováním databázových tabulek je mnohdy zlepšována výkonnost dotazů. Indexy jsou uchovávány informace o tom, kde jsou fyzicky uloženy záznamy v tabulce s konkrétními klíči. Můžeme ji rozdělit na indexování dimenzí a indexování faktové tabulky.

Indexování dimenzí je chápáno jako interaktivní aktivita, v dimensionálních tabulkách může být obsaženo mnoho indexů. Implicitně se vytváří jednoznačné indexy v rámci integritního omezení primárního klíče. Explicitně je doporučována indexace cizích klíčů jakožto atributů složeného primárního klíče v případě tabulky faktů. Po určité době provozu systému se řeší pokročilejší indexace na základě prováděných dotazů a logových souborů.

Dimenzionální tabulky jsou obvykle obsaženy více než jen jedním indexem. Rozlišujeme několik druhů indexů, například rozdělení podle jedinečnosti (Kimball a Ross, 2002), (Ballard, 2006), (Creating Indexes, 2015):

- *jednoznačný/unikátní (Unique)* – je jím zaručováno, že v klíčovém sloupci žádné dva řádky tabulky nejsou vyjadřovány stejnou duplicitní hodnotou, je tvořen z jednoho nebo více sloupců obsahujících unikátní klíč, unikátními indexy je zajišťována datová integrita na daném sloupci a je aplikována pro urychlení vyhledávání,
- *víceznačný (Non-unique)* - nejsou jím kladena omezení jedinečnosti na hodnoty sloupců, hodnotou indexu je ukazováno na umístění řádku.

Rozdělení založené na fyzické organizaci (Ballard, 2006), (Němec, 2014):

- *b-strom (B-Tree)* - listové uzly jsou dány hodnotou indexu a je jimi poukazováno na fyzický řádek, využíváme je na attributech s vysokou kardinalitou a unikátními hodnotami, je pokládán za vhodný pro data, která jsou často užívána v operacích spojování (join) nebo vyhledávána při využívání jen omezené skupiny záznamů,
- *bitmapové indexy* - každá jedinečná hodnota je zastoupena v bitmapě, využíváme je na všech attributech se střední a nízkou kardinalitou při vysokém počtu opakujících se hodnot.

Rozeznáváme i další druhy indexů, nicméně cíl diplomové práce není orientován na definování všech.

Faktovými tabulkami je prezentován speciální případ pro indexaci, jejich primární klíč je vytvářen téměř pokaždé složením podmnožiny cizích klíčů. V uvedeném případě využíváme jeden zřetězený index na primárních klíčích dimenzí ve faktové tabulce (Kimball a Ross, 2002).

Rozdělení tabulek (partitioning)

Zde se snažíme o efektivní rozdělení tabulek napříč několika disky, kdy data jedné logické tabulky jsou rozdělena napříč několika fyzickými (Larissa, 2003).

Rozdělení dat v tabulce na menší části je často realizováno podle řádků, sloupců nebo obojího. Pokud rozdělujeme podle sloupců, rozdělujeme vertikálně, podle řádků horizontálně. Rozdělováním velkých faktových tabulek je zvyšována výkonnost.

Obvykle je založeno rozdělování faktové tabulky na dimenzi času z dimenzionálního modelu, například podle roků nebo měsíců. Je pokládáno za vhodné rozdělit i velké dimenzionální tabulky (například tabulka zákazníků).

Přednost v rozdělení tabulek je viděna především v jednodušší restrukturalizaci, indexování, sekvenčním procházení, reorganizování, obnovitelnosti, monitoringu, flexibilním přístupu k datům a efektivnímu data managementu a škálovatelnosti (Ballard, 2006).

2.5.4) Návrh a vývoj ETL procesů a úloh

Vývojem procesů extrakce, transformace a nahrání (ETL) je uživateli skryta část pro většinu BI projektů. Účel fáze je spočíván v detailní definici transformačních pravidel mezi produkčními a analytickými daty s cílem zajistit dostatečnou kvalitu dat (Novotný, 2005).

Daný vývoj popisují Kimball a Ross (2013) v 10 krocích. Před začátkem tohoto procesu bychom měli mít hotový logický design, vypracovaný plán architektury a mapovány všechny datové elementy ze zdrojového do cílového systému (Kimball a Ross, 2013).

Nakreslit více-úrovňový (high-level) plán - proces návrhu začneme s velmi jednoduchým schématem zdrojů a cílů, který budeme postupně doplňovat. Plán není odkázán na konkrétní technologie a přístupy.

Výběr ETL nástroje - známe množství dostupných nástrojů pro ETL datové sklady. ETL nástroji jsou čtena data z řady zdrojů zahrnující databázové soubory, ODBC, OLE DB a další. Jsou obsahována funkcionality pro definování transformací dat a pro řízení celkové logiky ETL systému.

Vývoj výchozích strategií - s celkovým plánem požadavků od ETL můžeme vytvořit skupinu výchozích strategií pro běžné aktivity v ETL systému, zmíněné aktivity zahrnují:

- *návrh extrakce dat z hlavních zdrojových systémů* - určíme výchozí metodu pro extrakci dat z každého zdrojového systému,
- *archivaci extrahovaných a nezpracovaných dat*,
- *politiku datové kvality pro dimenze a fakta* - datovou kvalitu bychom měli monitorovat již během ETL procesu, pro kontrolu datové kvality můžeme provést tři druhy testů - sloupcový, test struktury a test obchodních pravidel.

Sloupcovým testem jsou testována data v jednom sloupci na přítomnost neočekávaných prázdných hodnot a hodnot mimo předepsaný rozsah nebo formát.

Testem struktury je ověřována relace dat mezi sloupci, jako skupiny vztahů 1:n, vztahy cizích a primárních klíčů a další.

Testy obchodních pravidel jsou řazeny mezi komplexnější testy, kterými je zkoumáno plnění nadefinovaných obchodních pravidel.

V případě zachycení problémů máme k dispozici tři řešení - zastavení procesu, zaslání chybného záznamu pro pozdější zpracování nebo označení chybných dat (Kimball a Ross, 2013).

- *definice postupů zpracování změn dimenzionálních atributů* - již byla probrána výše v rámci dimenzionálního modelování u určení SDC a RCD. V tomto bodě je realizován návrh fyzické implementace navrhnutého řešení.
- *zhodnocení stavu dostupnosti zdrojových a cílových systémů* - pro splnění požadavků na dostupnost je nutnost potřeby jejich dokumentace, především dokumentace dostupnosti jednotlivých datových zdrojů.
- *návrh subsystému pro audit dat* - každý řádek v databázové tabulce by měl být svázán s auditními informacemi, popisujícími jak data prochází systémem.
- *organizace dočasného úložiště pro ETL*.

Detailní analýza navrhovaného datového úložiště - zde více analyzujeme informace nezbytné pro naplnění každé tabulky cílového datového úložiště včetně profilování dat. Budeme se zabývat ujištěním čistých hierarchií, vývojem detailního schématu tabulek a vytvořením specifikačního dokumentu ETL.

Čistá hierarchie je představována tím, že žádná položka není zařazována do více jak jedné kategorie. V nejlepších zdrojových systémech jsou normalizovány úrovně hierarchie do několika tabulek, s omezením pomocí cizího klíče mezi jednotlivými úrovněmi. Pokud hierarchie není normalizována, je považováno za potřebné ji vyčistit (normalizovat) nebo uznat, že se nejedná o hierarchii (Kimball a Ross, 2013). Pokud bychom tento problém neřešili, pravděpodobně bude docházet k dvojímu započtení zůstatků z důvodu, že jednou entitou dítě bude reportováno více než jedné entitě rodič (Jethwa, 2007).

Specifikačním dokumentem ETL jsou shrnovány předcházející fáze vývoje ETL plánu, a to hlavně výchozí strategie (datová extrakce, archivace, sledování datové kvality, management změn dimenzionálních atributů), víceúrovňová schémata a návrh tabulek s detailním mapováním zdrojů a cílů (Kimball a Ross, 2013).

Plnění dimenzionálních tabulek historickými daty - plnění je odkázáno na typ dimenzionálních tabulek z pohledu historizace. Zabýváme se plněním dimenzionálních tabulek, dimenzionální transformací, kombinací několika datových zdrojů, potvrzením 1:n a 1:1 vztahů, přiřazením cizího klíče dimenzionálním tabulkám a plněním datumových a ostatních statických dimenzí.

Plnění dimenzionálních tabulek SCD typu 1 - jedná se o nejjednodušší typ dimenzionální tabulky pro plnění, je pouze přepisována historie. I přes to je brána v potaz potřeba provést dimenzionální transformaci ve formě čištění dat a přiřazení umělých klíčů. Skládá se z následujících kroků:

- jednoduché datové transformace - jedná se o konverzi datových typů, nahrazení prázdných hodnot výchozími hodnotami,
- kombinace různých oddělených zdrojů,
- dekódování produkčních kódů - týká se běžné spojovací úlohy při přípravě dat, snažíme se vyhledat ekvivalentní textové popisy produkčních kódů,
- validace vztahů mnoha k jedné a jedna k jedné,
- přiřazení umělých klíčů dimenzím - jakmile jsme si jistí, že dimenzionální tabulky jsou složeny z jednoho řádku pro každý jedinečný dimenzionální člen, můžeme přiřadit umělé klíče, v rámci ETL procesu spojíme identifikátory řádků z produkčního systému s umělými klíči,
- nahrání dat do dimenzionálních tabulek - nahrávací proces do cílových tabulek není považován za složitý a je uskutečňován, až jsou připravena všechna dimenzionální data.

Plnění dimenzionálních tabulek SCD typu 2 - jak bylo již definováno, jedná se o typ, kterým je uchovávána historie přidáním nového řádku do dimenzionální tabulky. Během prvního nahrání musíme znovu vytvořit historii pro takto spravované dimenzionální atributy z důvodu, že pokud uživatelé identifikují atributy jako důležité pro procházející historii, budou chtít procházet historii zpátky v čase i před vytvořením datového úložiště (Kimball a Ross, 2013).

Plnění faktových tabulek historickými daty - ve jmenované fázi provedeme transformaci dat do faktových tabulek, a ta se zabývá převážně nahrazením prázdných hodnot, pivotováním dat před vypočtením odvozených kalkulací a referenční integritou mezi faktovou tabulkou a dimenzemi, následně se provádí samostatné nahrání.

- Před prvním nahráním do historické faktové tabulky provedeme identifikaci jednotlivých záznamů pro extrakci, ještě jednou se ujistíme, že tyto záznamy jsou užitečné hlavně z obchodního pohledu.
- Auditní statistiky - při plnění faktových tabulek můžeme sledovat množství auditních informací. Během plánovací fáze návrhu ETL systému bychom měli identifikovat auditní informace jako metriky datové kvality. Těmito metrikami jsou představovány především kalkulace počtů a součtů, kterými můžeme provést křížovou kontrolu integrity a porovnat tak datové úložiště a zdrojový systém.
- Transformace faktové tabulky - nejčastěji je prováděna nahrazením prázdných hodnot, pivotováním dat a před počítáním odvozených kalkulací.

Před nahráním tabulky faktů se musíme ujistit, že každý klíč historických faktových dat z produkčního systému je propojen s dimenzionální tabulkou. Propojení se realizuje pomocí umělého klíče, který vytvoříme na mapování klíčů z produkčních systémů za využití ETL nástroje „lookup operátora“ nebo SQL skriptu, kterým je daná akce provedena. Po propojení přirozených klíčů s umělými jsou připraveny řádky faktové tabulky na nahrání, které následně provedeme.

V ***plnění dimenzionálních tabulek přírůstkovými daty*** vidíme podobnost s předchozí zmíněnou fází s historickými daty s rozdílem, že je potřeba identifikovat nové, změněné a smazané řádky. Skládá se z následujících třech částí – extrakce dimenzionálních tabulek, identifikace nových a změněných řádků a procesu změny dimenzionálních atributů.

Extrakce dimenzionálních tabulek - za nejjednodušší je pokládáno vytvoření současného snímku dimenzionálních tabulek a nechat transformaci, aby jí bylo určeno, co bylo změněno a

jak se s tím vypořádat. Tím však riskujeme velké časové ztráty v tabulkách. Máme-li možnost, měli bychom extrahovat pouze změněné řádky.

Identifikace nových a změněných řádků - data z produkčního systému jsou identifikována svým přirozeným klíčem, který musí být propojen ke stejnému řádku v aktuální dimenzi. Nové záznamy tak najdeme porovnáním produkčních klíčů mezi tabulkou relačního zdrojového systému a dimenzí. Všechny řádky, u kterých není shledána shoda, jsou brány za nové řádky a tyto řádky posléze nahrajeme do dimenzionální tabulky. Dalším krokem je porovnávání, zda byly změněny řádky, za nejjednodušší techniku se chápe porovnání řádku po řádku mezi zdrojovými daty a dimenzemi.

Proces změny dimenzionálních atributů - prvním krokem je rozhodováno, zda daný řádek je již obsažen v dimenzi, pokud ano, jestli byl změněn atribut. Změny aplikujeme na základě toho, jak jsme definovali SCD pro danou tabulku. Využijeme následující workflow, v první fázi zjistíme, zda se jedná o nový řádek, pokud ano, zapíšeme jej do dimenze. Ve druhé fázi zjišťujeme, zda byly změněny nějaké řádky, pokud ano, rozhodnutí o další akci je odkázáno na SCD typ.

Plnění faktové tabulky přírůstkovými daty - oproti práci s historickými daty musí být tento proces plně automatizován, jakmile jsou extrahovány nové a změněné řádky ze zdrojového systému, je ověřována datová kvalita a následně nahrána do faktových tabulek datového skladu.

Extrakce faktové tabulky a bod kontroly datové kvality - jakmile jsou extrahovány nové a změněné záznamy ze zdrojového systému, měly by být uloženy do dočasného úložiště. Tam jsou uloženy z důvodu možnosti auditu a pro jejich transformaci bez zatěžování zdrojového systému.

V transformaci faktové tabulky a přiřazení umělého klíče shledáváme podobnost s historickým nahráváním. Rozdíl nacházíme v automatickém zvládnutí při porušování referenční integrity. Toho dosáhneme pomocí několika metod:

- zastavením nahrávání,
- zahazením chybných řádků,
- uložením chybných řádků do souboru nebo tabulky pro pozdější analýzu,
- opravou chyb pomocí vytvoření řádku okrajové dimenze a vrácením umělého klíče,
- opravou chybného řádku mapováním na jeden speciální řádek „neznámí“ v každé dimenzi.

Pro většinu systémů vyhledáváme umělý klíč odkazem na dotaz, pohled nebo fyzickou tabulku obsahující podmnožinu dimenzionální tabulky (Kimball a Ross, 2013).

Pozdě přichází fakta - vznikají, pokud využíváme například distribuované datové zdroje napříč několika servery a současně se vyskytují problémy s internetovým připojením nebo latencí. Pokud využíváme SCD typu 1 – přepsání, pozdě přichází fakta nejsou představována žádným problémem, v případě využití vyšších SCD typů musíme daná fakta propojit s členem dimenze, který byl ovlivněn při vzniku faktu. To se řeší nejčastěji datem začátku a konce účinnosti v rámci dimenze.

Agregace dat v tabulkách a nahrání dat do OLAP databáze - agregované tabulky jsou představovány výsledky agregací hodnot uložených ve faktových tabulkách. Je zde možnost výskytu následující výzvy, pokud jsou obsaženy v agregované faktové tabulce dimenzionální atributy, kterými je využíván SCD typ 1 (přepisování atributů). Každou změnou podobného dimenzionálního atributu jsou ovlivňovány všechny agregace faktové tabulky a OLAP kostky využívající daný atribut.

Musíme proto zajistit, aby agregace byly synchronizovány s původní faktovou tabulkou a nedocházelo tak k rozdílným výsledkům na základě dotazů do detailních faktů a do předem počítané agregované faktové tabulky.

Automatizace fungování ETL systému - v ideálním případě se spouští ETL systém automaticky a pravidelně, je jím prováděna extrakce, transformace a nahrání dat bez potřeby lidského zapojení. Tímto bodem je i zajišťováno automatické řešení předpověditelných výjimek a chyb a vhodné ošetření nepředpověditelných a zcela náhodných chyb.

Plánování úloh je obvykle pojímáno přímočaře, v ETL nástrojích jsou obsaženy funkce pro naplánování spouštění úloh v daný čas, provedení následných úkolů v případě úspěšného dokončení a sledování, zda nebyly vytvořeny nové záznamy během nahrávání. Údaje o průběhu ETL procesu mohou být zaznamenávány průběžně do auditní dimenze.

Za automatizaci zvládání předpověditelných výjimek a chyb je považováno například porušení referenční integrity a obdržení nulových hodnot. Chyby zachytíme a opravíme v rámci ETL systému. Některé chyby není možno předpovědět, například zkomolená data, výpadek elektrického proudu během zpracování a podobně. Tyto chyby jsou opravovány manuálně, a proto se snažíme zajistit co nejlepší podmínky pro obnovu bezchybnosti (Kimball a Ross, 2013).

Tímto byla ukončena podkapitola zabývající se návrhem ETL systému, ve které jsme se zabývali návrhem ETL plánu datových pump pro historické nahrání dat, následně návrhem datových pump pro přírůstkové nahrávání dat a v závěru agregací, OLAP a automatizací celého ETL systému.

Současně je zakončena i teoretická část diplomové práce, v níž byl nastíněn základní úvod do terminologie BI, představen princip multidimenzionality, definovány jednotlivé komponenty systému BI a jejich vzájemné propojení v rámci architektury BI řešení. Zabývali jsme se Kimballovým procesem životního cyklu business intelligence, především analýzou požadavků, dimenzionálním modelováním, fyzickým návrhem a návrhem ETL.

Uvedené metodologické postupy, zejména analýzu, dimenzionální modelování a návrh ETL, budou nyní prakticky ověřeny při aplikaci na konkrétní problém ve výrobní společnosti. Výše zmíněnou aplikací se bude zabývat další část diplomové práce.

3) Analýza stavu řešení řízení IT služeb

Analýzu současného stavu započneme základními informacemi o společnosti, následně přejdeme na podrobnou analýzu současných procesů a způsobu řízení IT služeb.

3.1) Úvod o společnosti

Mezinárodní společnost s 27 000 zaměstnanci působící ve více než 35 zemích je veřejně obchodovatelná a zapsána na burzách NASDAQ OMX v Helsinkách a Stokholmu. Prodeje společnosti dosáhli v roce 2014 částky 10,2 miliardy eur a čistý zisk před zdaněním částky 810 miliónů.

V roce 1888 je zaznamenáván počátek historie společnosti Stora Kopparbergs Bergslags a od roku 1984 je používán zkrácený název Stora. V roce 1889 byla založena společnost Enso Trasliperi Aktiebolaget, od roku 1996 používající název Enso Oy. Začátkem roku 1998 byla vybudována nová společnost Stora Enso Oyj, a to spojením těchto dvou výše uvedených společností. V celé historii firmy bylo hojně využíváno akvizicí a joint venture k pronikání na nové trhy a upevnění pozice na současných trzích.

Společnost se zabývá výrobou papíru, bio materiálů, dřevěných produktů a obalovým průmyslem. Její zákazníci jsou tvořeni vydavateli, tiskárnami a obchodníky s papírem.

Moto společnosti je formulováno následně - „konat dobro pro lidi a planetu“. Motu je uzpůsobována i firemní strategie ve směru obnovitelných zdrojů. Firemní hodnoty jsou představovány jako „Vedení, Dělat co je správné“, kdy společnost myslí na dopady svých budoucích činností na okolí a vedení ve smyslu jasně definovaného cíle.

Soustřeďuje se na evropské trhy, v současné době zejména na rostoucí trhy v Asii a Latinské Americe. Je rozdělena organizačně podle druhu produkce na pět divizí – consumer board, biomaterials, wood products, paper a packaging solutions. Kromě zmiňovaného rozdělení disponuje podpůrnými a zásobovacími útvary. Nyní něco konkrétněji k obchodním oblastem společnosti.

Biomaterials - soustřeďuje se na výrobky z vlákniny, které splňují požadavky na papír, lepenku a požadavky tkáňových producentů. Vyrábí vlákninu z obnovitelných zdrojů. Posláním této divize je najít nové inovativní způsoby pro efektivní využití dřeva při současné vedlejší produkci vlákniny.

Wood products - poskytuje inovace založené na dřevěných materiálech a řešení pro každodenní potřeby života a bydlení. Produkty divize pokrývají všechny oblasti městské

výstavby od nosné struktury po vnitřní design a environmentální stavby. K dalším produktům jsou řazeny masivní dřevěné prvky, stavební moduly, dřevěné komponenty, kuličky a různá řeziva. Zákazníci pochází jak z veřejného, tak soukromého sektoru a jsou reprezentováni stavebními firmami, obchodníky a prodejci.

Paper - zajišťuje obnovitelný papír pro tisková média a kancelářské použití. Jejich zákazníci jsou vydavatelé, tiskárny, maloobchodníci, velkoobchodníci, dodavatelé kancelářských potřeb. Sortiment zahrnuje novinový, knihový papír a časopisový papír, křídový papír a kancelářský papír.

Packaging solutions - divize se soustřeďuje na obalové materiály založené na vláknu a inovačním balení pro zboží zákazníků a průmyslovou aplikaci. Důraz je kladen na obnovitelné zdroje, které se využívají v celém hodnotovém řetězci od výroby buničiny k výrobě materiálu a obalů až po opětovnou recyklaci. Jako příklady produktů si uvedeme grafické desky a balicí materiál, kromě toho jsou vyráběny obaly z vlnitého materiálu, jako jsou transportní obaly, krabice, papírové sáčky, pytlíky a další.

Consumer board division - nejnovější divize byla oddělena od Packaging solutions, poskytující vysoce kvalitní obaly pro tištění a balení. Obaly jsou využívány pro balení tekutin, potravin, léků, cigaret, čokolády, kosmetiky a luxusních výrobků.

Společnost expanduje do rostoucích trhů jako je Čína, Indie a Pákistán. V současnosti se nachází výrobní závody ve Finsku, Švédsku, Španělsku a ve východní a centrální Evropě.

Kromě těchto divizí je vlastněna společností ještě jedna nevýrobní podpůrná divize zaměřená na energii, logistiku, obchodní informační procesy (BIS) a operace dodávky dřeva ve Švédsku a ve Finsku (Stora Enso, 2015).

V České republice jsou provozovány dvě pobočky společnosti, a to v Ostravě a v Žďarce nad Doubravou. V Ostravě můžeme navštívit sídlo SDCC Ostrava (Software Development Competence Centre), jež bylo založeno v roce 2006 pro interní vývoj softwaru pro mateřskou společnost Stora Enso. Zabývá se vývojem plánovacích, logistických a prodejních systémů převážně pro divizi wood products a paper. Jedná se o největší IT centrum společnosti zaměstnávající více než 120 zaměstnanců. Rozděluje se na BIS (Business Information Services) a RAACO (Reporting & Analysis Applications and Collaboration Team). Pod hlavičkou tohoto centra je vypracována tato diplomová práce (Stora Enso, 2013).

3.2) Analýza podnikových systémů

3.2.1) Zadání práce

Před samotnou analýzou jednotlivých podnikových systémů je považováno za nezbytné přesně definovat požadovaný výstup. Tato definice byla provedena pomocí rozhovorů se zadavatelem práce – zákazníkem a konzultantem. V našem případě obě osoby pracují pro stejný podnik, konzultant v oddělení Microsoft BI (MSBI) a zákazník – společnost, je zastoupena v tomto případě manažerem odpovědným za bezproblémový chod IT procesů společnosti a jejich další rozvoj.

Na základě konzultace se zákazníkem byl vytvořen následující obecný požadavek – optimalizovat současný způsob reportingu výkonnosti podnikových IT procesů s důrazem na vypovídající grafické zpracování a především automatizaci tvorby reportů.

Následně byl uskutečněn rozhovor s konzultantem pro zjištění konkrétních nezbytných kroků neboli byla zjišťována definice pro zadání technické stránky řešení. V rámci tohoto rozhovoru byla definována následující specifikace a nástin postupu prací:

- je potřeba analyzovat, jaké IT procesy jsou využívány společností,
- ke každému IT procesu nalézt, které systémy a aplikace jsou používány,
- zjistit vlastníky těchto systémů a aplikací,
- zjistit současné nastavení KPI u těchto systémů,
- provést konzultace s vlastníky aplikací pro zjištění, zda deklarované KPI byly naimplementovány a pomocí rozhovoru zjistit, zda by bylo přínosem zavést nové KPI,
- zahájit samotnou analýzu konkrétních systémů a jejich datových zdrojů,
- navrhnout vhodnou architekturu datového skladu (DW) pro reporting,
- navrhnout procesy extrakce, transformace a nahrání dat do vytvořeného datového skladu,
- navrhnout datové kostky nad vytvořeným datovým skladem pro podporu interaktivního reportingu,
- navrhnout interaktivní reporty založené na datových kostkách v produktu Microsoft Excel.

K této práci byla současně poskytnuta následující SW podpora – Microsoft SQL server 2013, Oracle SQL Developer a Microsoft Visual studio.

3.2.2) Analýza podnikových procesů

Pro analýzu jednotlivých IT procesů byl poskytnut dokument „IT Service Level Agreement“, který byl zpracován v roce 2014 externí společností pro danou firmu a je možno ho považovat za poměrně aktuální. Výstupy prvotní fáze analýzy jsou shrnovány v dokumentu, obsahujícím používané aplikace společnosti, odpovědné osoby a část KPI. Pro ukázkou dokumentu je přiložena výšeč několika aplikací – obrázek 3-1. Celý dokument zahrnuje přibližně 90 služeb/procesů (service) každé ze čtyř divizí společnosti². Po první analýze finální verze dokumentu bylo zjištěno, že přibližně 94% služeb můžeme považovat napříč divizemi za stejné, pouze divize wood products je více odlišována od ostatních v používaných systémech. Z toho bylo vyplýváno, že k další analýze bude nezbytně podstoupeno nejméně 90 procesů.

Service group	Service sub-group/ service	Owner	Systems	KPI
Global Platforms	Network Services	Kimmo Kainulainen, Andres Köster		Number of Network connections, Traffic reports
Global Platforms	Remote Access	Peter Åberg		Availability, Number of users, Number of concurrent users
Global Platforms	External Connections DataCenter	Juha Salminen		Request certificate (3 business days); Number of connections; Number of users; Number of Active Firewall Rules
Global Platforms	External Connections Regional	Anne Komscha		Certificate Delivery (3 business days); Number of connections; Number of users; Number of Active Firewall Rules
Global Platforms	Internet Access/Web content filtering	Kimmo Kainulainen, Andres Köster		Availability, Number of users, Traffic Volume Amount of data (Amount of traffic exchanged), Amount of blocked requests
Global Platforms	Hosting Services	Sami Sipponen		Service availability, Number of NOB, ADT, MDT cases per invoice month, Volumes (reported as part of invoices)
	Database Services	Petr Holusa		N/A

Obr. 3-1: Služby společnosti

Zdroj: (vlastní zpracování)

V následujícím kroku byl konzultován vytvořený dokument s IT kontrolorem společnosti, který poskytl část dodatečných informací a poukázal na změny současného stavu oproti deklarovaným službám a KPI. Zbýlé chybějící a neověřené informace bylo nezbytné zkonfrontovat s realitou.

Po dokončení této fáze bylo přikročeno k jednotlivým rozhovorům s vlastníky služeb pro ověření KPI. Výsledkem bylo zjištěno, že část KPI je implementována v původním rozsahu a měří se, nicméně velká většina je buď implementována v modifikaci, nebo byla již zrušena a je nahrazena jinými KPI. Současně bylo zjištěno, že pracovníci jedné divize mají přehled o fungování systému pouze v jejich divizi, nikoli v ostatních divizích, spravovaných zpravidla z jiných IT center společnosti. Na základě těchto nově zjištěných skutečností byla provedena úprava vytvořeného dokumentu.

Aktualizovaný dokument byl předložen k prodiskutování s konzultantem a zákazníkem. Za výsledek můžeme označit shodu na nutnosti snížit původně plánovaný rozsah projektu na menší

² Pátá divize je výsledkem vzniku po provedení této analýzy rozdělením jedné divize na dvě a přejmenováním ostatních divizí. V práci je již používáno nové označení divizí platné od 1.1.2015.

část, na jednu divizi společnosti, konkrétně wood products. Za hlavní důvod je pokládán velký rozsah služeb a časově náročnější komunikace s množstvím vlastníků služeb napříč společností.

Výše uvedená divize byla vybrána především z důvodu zaměření oddělení MSBI právě na tuto divizi a znalosti části procesů z ostravského centra.

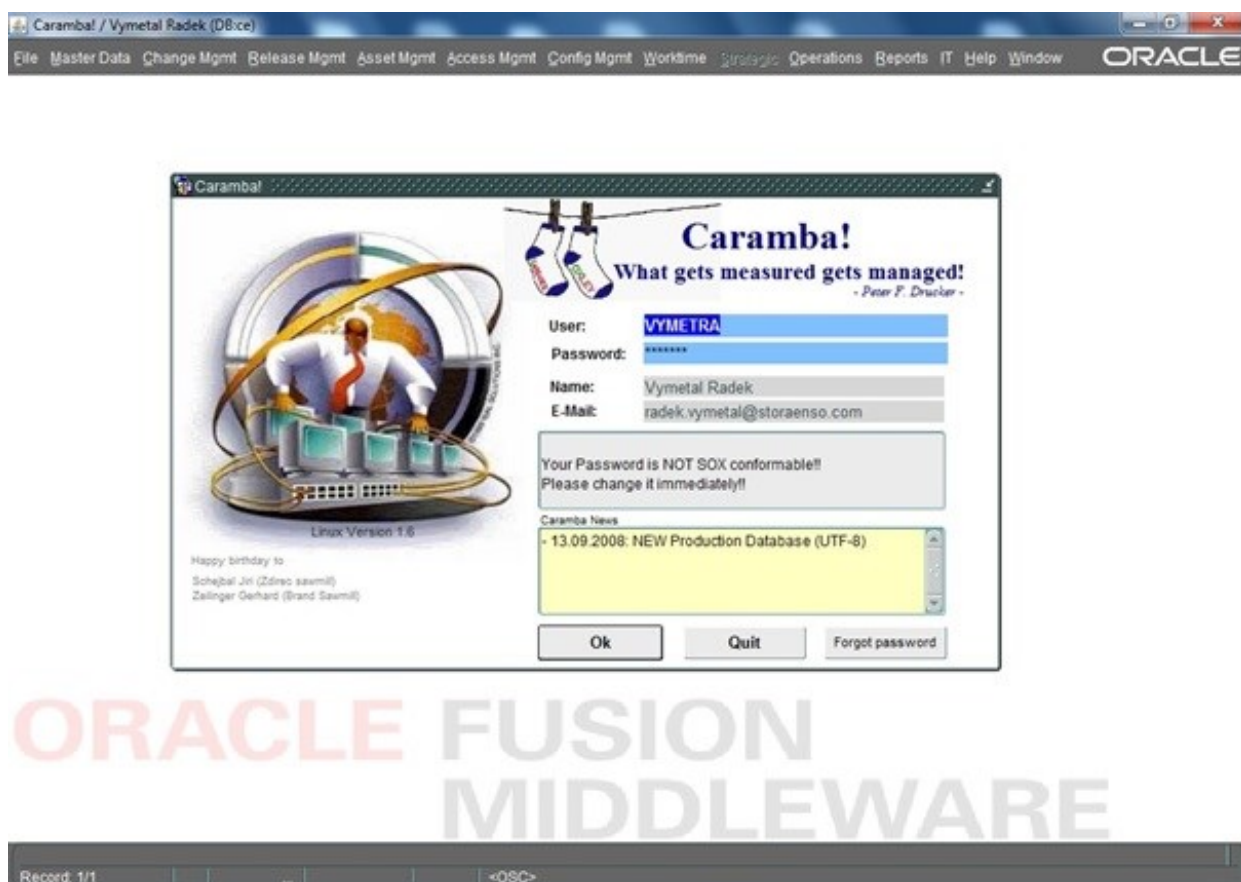
Po podrobnější analýze divize wood products bylo zjištěno, že část procesů bude modifikována a budou vytvářeny nové KPI. Nicméně procesem bude vyžadována delší doba, v řádu několika měsíců, proto není možno vyčkávat na provedení modifikací. Z toho důvodu se diplomová práce dále zaměřuje jen na část služeb, které se nemění. Konkrétně půjde o níže uvedené služby web page services (WPS), eSolutions a Caramba.

WPS - jedná se o službu zajišťující intranet, internet a online objednávkový systém. V rámci KPI se pro ni měří počet přístupů na webové stránky a jejich průměrná doba dle jednotlivých severů a vytíženost jednotlivých serverů. Veškeré měření je prováděno na základě textových logů jednotlivých serverů, kdy z logů jsou vybrány užitečné informace. Informace jsou následně konsolidovány a zobrazeny uživatelům v grafické podobě v rámci intranetu. Po bližším prozkoumání bylo zjištěno, že veškeré užitečné informace jsou již z logu vytaženy a k detailnějším informacím z důvodu ochrany údajů není možné přistupovat. Z toho důvodu systém hodnotíme jako nevhodný pro další analýzu.

eSolution - skládá se ze služeb eTimber Japan, ePellets, eAustralia, eBuild. Po podrobnější analýze této služby bylo zjištěno, že reporting se již provádí s vysokou vypovídající schopností a zahrnutím většiny dostupných dat. Bylo by možné zlepšení pouze s minimálním přínosem pro společnost.

Caramba - je systémem postaveným na platformě Oracle. Je používán jako datová základna a statický reportovací systém pro velké množství kontrolních procesů, jako pracovní výkazy, evidence stavu rozpracovanosti úkolů, rozdělování a čerpání rozpočtu, evidence zaměstnanců, požadavků pro nákupy nových pracovních prostředků a mnoho dalšího.

Vzhledem k zaměření práce pro optimálnější měření KPI jednotlivých procesů je systém Caramba vnímán jako vhodný systém pro tuto práci. Dále se zaměříme na právě zmíněný systém. Grafické rozhraní systému můžeme vidět na obrázku číslo 3-2 Vstupní rozhraní systému Caramba.



Obr. 3-2: Vstupní rozhraní systému Caramba

Zdroj: (vytvořeno reportovacím systémem Caramba)

Caramba je považován za grafický reportovací a obslužný systém obsahující správu a reporting důležitých IT služeb. Caramba je současně představována i relační databází v prostředí Oracle, stojící na pozadí. Grafickým rozhraním jsou zahrnovány následující oblasti:

- *master data* - jsou upravovány a zobrazovány osobní profily jednotlivých pracovníků, distribuční listy, uživatelské skupiny, služby (service), databáze znalostí řešení IT událostí a další záležitosti týkající se master dat,
- *management změn* (Change management) - jeho pomocí je pokrýván celý životní cyklus RfC požadavků, přes jejich registraci, schválení, vývoj a testování,
- *release management* - je v něm obsažena registrace, plánování, implementace a seznam vydaných verzí jednotlivých programů,
- *management aktiv* - požadavky na objednávky, schválené objednávky, dodané a skladované aktivity,
- *management přístupů* - požadavky pro přístup, schválený přístup a seznam všech schválených přístupů,
- *konfigurační management* - pro nastavení jednotlivých služeb,

- *plánování pracovní doby* (Worktime) - registrace, plánování pracovní doby včetně stavu plnění úkolů,
- *operace* - obsahující centrum znalostí,
- *reporty* - shrnující předchozí oblasti.

Z provedené analýzy je vidno, že v systému je zahrnuto hned několik procesů společnosti. Pro podrobnou analýzu systému, především jeho datové stránky v podobě relační databáze, budeme využívat software Oracle SQL Developer a Oracle SQL Data Modeler.

Oracle SQL Developer využíváme k prozkoumávání jednotlivých tabulek a hledání relevantních informací vhodných pro následné nahrání do datového skladu. Naopak pomocí Oracle SQL Data Modeleru zjišťujeme aktuální nastavení schématu databáze a jednotlivých vazeb mezi tabulkami, tedy využíváme reverzního inženýrství z důvodu chybějící softwarové dokumentace.

Samotný projekt vytvoření BI řešení bude realizován softwarem společnosti Microsoft. Konkrétně budou brány v úvahu produkty SQL Server 2012, Visual Studio a k nim přidružené nástroje.

V této kapitole byly nastíněny základní informace o společnosti, zabývali jsme se požadavky společnosti kladené na diplomovou práci, především na nový výsledný reportovací systém. Následně jsme se zabývali hloubkovou analýzou podnikových procesů s cílem nalézt vhodný proces pro splnění požadavků společnosti, výsledkem fáze byly zjištěny velké rozmanitosti procesů ve společnosti. V další fázi jsme zvolili jeden podnikový systém Caramba, která je hodnocena jako ideální kandidát pro splnění požadavků společnosti. Tento systém byl podrobněji analyzován ke konci kapitoly.

V následujících kapitolách se budeme zabývat návrhem řešení pro zefektivnění řízení služeb pomocí nástrojů business intelligence, kdy hlavní oblastí našeho zájmu bude systém Caramba.

4) Návrh řešení pro zefektivnění řízení služeb pomocí nástrojů business intelligence

V předchozí části práce byly definovány podnikové požadavky pro diplomovou práci. Nyní se budeme zabývat dalšími kroky Kimballova procesu budování systému BI, především dimenzionálním modelováním, fyzickým návrhem, návrhem ETL a celkovou implementací řešení. Kromě Kimballova procesu budou vytvořeny datové kostky a provedeno testování reportingu.

4.1) Dimenzionální modelování

V dimenzionálním modelování budeme postupovat podle Kimballova procesu dimenzionálního modelování, popsaného v teoretické části práce. Proces bude složen ze čtyř hlavních kroků – výběru podnikového procesu, deklarace podrobnosti informací o podnikovém procesu, identifikace dimenzí a identifikace a výběru numerických faktů.

4.1.1) Výběr podnikového procesu

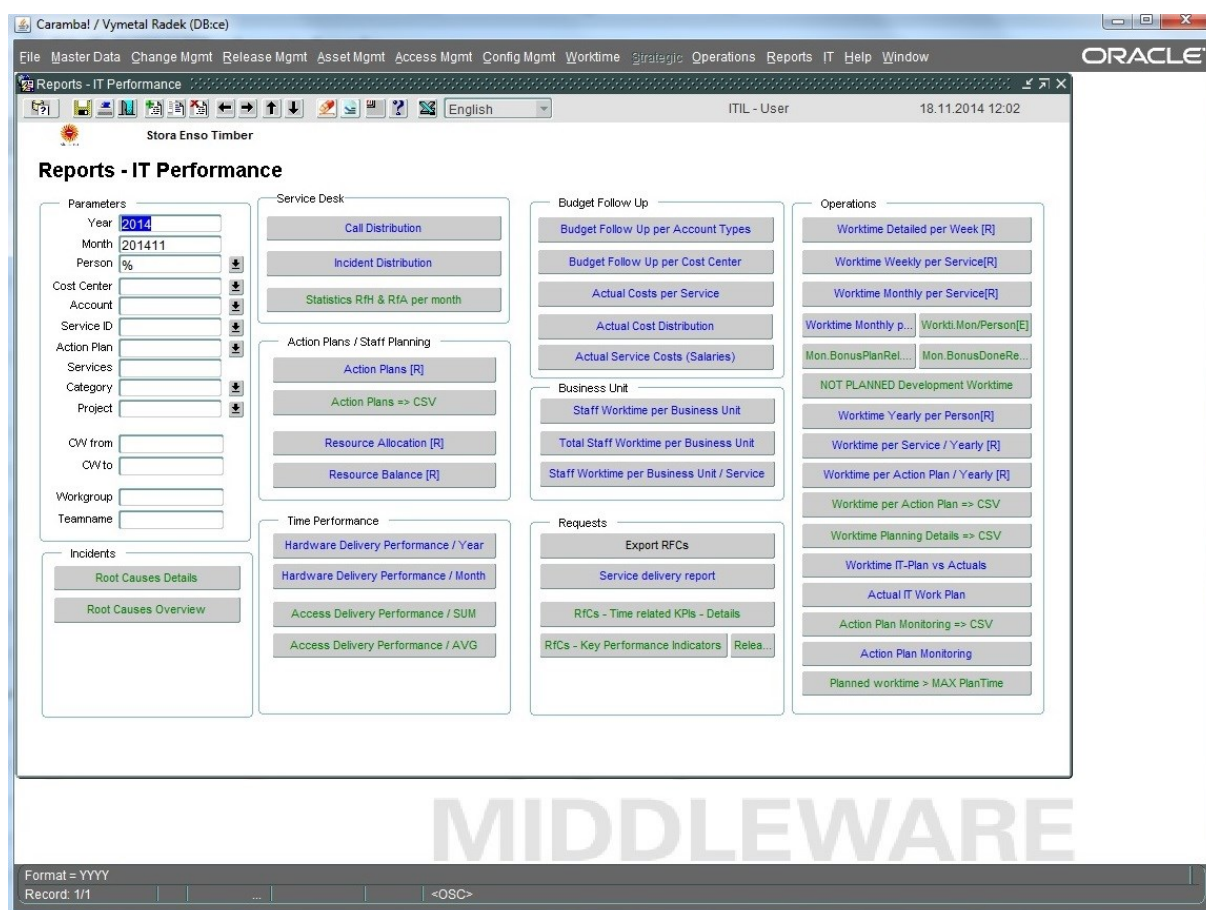
Vytvoření celopodnikového seznamu procesů

Celopodnikový seznam procesů již byl vytvořen v předcházejí podkapitole číslo 3.2.2. Zde se zabýváme jen podprocesy zachycenými v rámci reportovací sekce nástroje Caramba, zobrazeným na obrázku číslo 4-1.

Současný reporting se provádí pomocí zadání vstupních parametrů, jako datum, osoba, nákladové centrum, účet, služba, akční plán, kategorie, projekt a další. Následně uživatel klikne na požadovanou oblast a na základě vstupních požadavků dojde k vygenerování souboru aplikace Microsoft Excel nebo dokumentu PDF.

V rámci nástroje můžeme identifikovat následující pohledy reportingu:

- správy incidentů,
- service desku,
- akčního plánu/ plánování zaměstnanců,
- časové výkonosti (time performance),
- plánování rozpočtu,
- obchodních jednotek,
- správy požadavků,
- operací.



Obr. 4-1: Reportovací část systému Caramba
Zdroj: (vytvořeno reportovacím systémem Caramba)

Po podrobnější analýze jsme rozeznali procesy plánování rozpočtu, plánování zaměstnanců, management incidentů, dodávky hardware a management přístupů. Na daném základě byla provedena konzultace se společností a následně bylo dohodnuto, že se těmito procesy budeme zabývat jakožto podprocesy řízení IT služeb bez konkrétního zaměření na jeden z procesů.

Identifikace podnikových procesů

Na základě teorie by měla být dalším krokem provedena identifikace nejvhodnějšího procesu z hlediska komplexnosti, dostupnosti dat, datové kvality a strategického obchodního významu. S ohledem na zadání společnosti a předchozí fáze, kdy byl vybrán konkrétní proces, jsou nyní pouze shrnována zmíněná kritéria.

Komplexnost analyzovaných procesů je udávána jako vysoká, dostupnost informací o výše uvedených podprocesech je požadována za velmi dobrou, veškeré informace jsou uloženy v rozsáhlé databázi na platformě Oracle. Z pohledu kvality dat se jedná o výstupy aplikace Caramba a automaticky generované identifikační kódy, riziko vzniku problémů s datovou

kvalitou nastává v případě textových uživatelských vstupů přes tuto aplikaci. Za vysokou shledáváme strategickou významnost podprocesů jako celku.

Více k jednotlivým podprocesům najdeme v příloze číslo 1 – BPMN procesní modely, kde jsou uvedeny výše zmiňované podprocesy - plánování rozpočtu, plánování zaměstnanců, management incidentů, dodávky hardware a management přístupů, jsou analyzovány graficky z pohledu podnikového workflow.

Identifikace entit a metrik

V této části identifikujeme entity sdílené napříč jednotlivými podprocesy, a to pomocí tabulky číslo 4-1, kterou jsou shrnovány jednotlivé možnosti.

Entity ► Podnikové procesy ↓	Entity ►								
	ActionPlan (Akční plán)	Users (Uživatelé)	Accounts (Účty)	Services (Služby)	Sites (Pobočky)	Project (Projekt)	Change management (Požadavky na změnu)	Hardware	Cost Centers (Nákladová centra)
Incident management (management incidentů)	X	X		X	X		X		
Hardware Delivery (dodávky hardware)		X	X		X			X	X
Staff Planning (plánování zaměstnanců)	X	X	X	X	X	X			X
Access Delivery (management přístupů)		X		X	X		X		
Budget Follow Up (plánování rozpočtu)	X	X	X	X	X	X			X

Tab. 4-1: Entity sdílené napříč jednotlivými procesy
Zdroj: (vlastní zpracování)

Vzhledem k počtu sdílených entit bude pravděpodobně vhodné jednotlivé podprocesy zahrnout do jednoho dimenzionálního modelu s více tabulkami faktů.

Identifikace zdrojů dat

V této části popíšeme zdroj dat pro vybraný podnikový proces – IT performance. Jedná se o centrální relační databázi uloženou na platformě Oracle. Přírůstky dat vychází z činnosti transakčních systémů a jsou postaveny na denní bázi. V otázce vlastnictví se jedná o interní data společnosti, která jsou vytvářena napříč podnikovými jednotkami.

Na základě analýzy grafického schématu databáze vytvořeného pomocí SQL Data Modeleru bylo zjištěno, že relační databáze obsahuje velké množství vzájemných vztahů mezi celkem 145 tabulkami.

Společností je kladen požadavek na převedení veškerých dalších využívaných dat z databáze do prostředí Microsoft SQL Serveru. Jako důvod je předkládáno zamezení možného negativního vlivu na výkon zdrojové databáze během vývoje BI řešení. Toto převedení bylo provedeno v první fázi pomocí DDL skriptu, vygenerovaného z SQL Data Modeleru.

Při převodu se objevovaly problémy s datovými typy obsahující velké množství dat, jako vložené PDF soubory a další. Druhý problém byl dán názvovými konvencemi z prostředí Oracle, kde je definováno psaní názvů tabulek a sloupců velkými písmeny, naopak v MS SQL Serveru se uplatňuje konvence psaní velkého písmena v názvech jako oddělení nových slov. Pro odstranění vzniklých problémů byla provedena manuální úprava vygenerovaného DDL skriptu, zejména úprava názvové konvence a zvolení vhodných datových typů. Po vytvoření databázových tabulek bylo realizováno nahrání dat ze zdrojového systému, a to pomocí procedur vytvořených na základě šablony uvedené v příloze číslo 2 - šablona procedury pro nahrávání dat.

Výběr přístupu ke sběru požadavků

Výběr přístupu ke sběru požadavků považujeme za velmi důležitý krok, v našem případě se jedná o částečně kombinovaný přístup založený jak na základní definici z pohledu uživatelů, tak především na dostupných datech a analýze E/R modelu zdrojových dat.

V rámci diskuze s kontaktními osobami ve společnosti byl vybrán preferovaný přístup založený na datech. Důvodem vývěru je velké vytížení zaměstnanců spojených s prací na modifikacích nastavení a KPI ostatních procesů. Rozhodnutím budou velmi ovlivněny další fáze, kdy budoucí uživatelé se budou zapojovat do vývoje pouze minimálně.

Získávání a analýza požadavků

Na základě předchozích fází budeme vycházet ze základních podnikových požadavků - obsáhnout současnou funkčnost reportovacího systému Caramba. Opíráme se zejména o podrobnou analýzu grafického rozhraní Caramba a o analýzu relační zdrojové databáze za pomoci zpětného inženýrství a jím vytvořeného E/R modelu.

Na základě uvedených analýz byly zjištěny následující požadavky uvedené v tabulce 4-2, které byly ohodnoceny prioritním stupněm od 1-5, kdy stupeň 1 je představován nejnižší prioritou a 5 naopak nejvyšší prioritou.

Pořadové číslo	Požadavek	Měřitelné hodnoty (ukazatele)	Úhly pohledu (dimenze)	Priorita
Q1	sledování incidentů a jejich příčin	počet incidentů a jejich příčin	RfC kategorie, služba, čas	5
Q2	sledování RfH statistik	počet objednaného HW	čas, pobočka	3
Q3	sledování RfA statistik	počet požadavků o přístup	čas, pobočka	3
Q4	přehled plánů	přehled plánů	čas, služba, akční plán, náklady	4
Q5	alokace zdrojů	přehled dostupných a alokovaných zdrojů	zaměstnanci, služba, čas	4
Q6	výkonost dodávek HW	počet objednaných a schválených HW	hardware, čas	4
Q7	sledování přístupové doby služeb	čas ke schválení přístupu a k aktivaci přístupu	služba, čas	3
Q8	sledování plnění rozpočtu na základě účtů, nákladových center, služeb	sledování nákladů (odhad, aktuální, odchylka)	účty, služba, nákladové centrum, čas	5
Q9	pracovní doba zaměstnanců v závislosti na obchodních jednotkách, službách, plánech	délka pracovních úkolů a náklady na zaměstnance	služba, akční plán, zaměstnanci, čas	5
Q10	stav vyřizování požadavků	RfC status	služba, RfC kategorie	5

Tab. 4-2: Seznam požadavků

Zdroj: (vlastní zpracování)

Z pohledu důležitosti jednotlivých požadavků se jeví společnosti jako klíčové všechny, nejdůležitější jsou pak viděny v sledování incidentů, rozpočtu, práci s požadavky a přehledu činností pracovníků. Jednotlivé požadavky jsou sledovány na základě reportů v následujících časových pohledech:

Q1 – detailní sledování na jednotlivých incidentech + sumarizace za měsíce,

Q2, Q3 – sledování za týden a měsíc,

Q4, Q5, Q8, Q9 – sledování za měsíce, roky,

Q6, Q7 – sledování za den, měsíc, rok,

Q10 – sledování za požadavek, měsíc a rok.

Vzhledem k rozmanitosti modelu a plánu využít datové kostky je považován za vhodnější způsob všechny uvedené metriky sledovat na začátku za jednotlivé požadavky nebo dny. Následně provedeme jejich sumarizaci v rámci kostek.

V našem případě historizací údajů musí být průběžně odráženy změny stavů jednotlivých požadavků a incidentů, tedy pravděpodobných faktů, kdy převážnou většinou entit je vyžadováno přepisování historie.

4.1.2) Deklarace podrobnosti informací o podnikovém procesu

Budeme se zabývat určením možných granularit faktové tabulky, neboli počtem úrovní podrobností modelovaného procesu.

Granularita faktové tabulky

Granularita odkazuje na úroveň detailu nebo sumarizaci jednotek dat v datovém skladu.

Fakta	1 řádek	Žádoucí úroveň granularity faktů
Plánování pracovní síly	1 řádek = pracovní plán na zaměstnance	Pracovní doba a náklady na pracovníka v závislosti na akčním plánu, projektech, službách, pobočkách, účtech a nákladových centrech.
Žádosti o přístup	1 řádek = 1 požadavek na přístup	Žádost o přístup ke službě, v závislosti na pracovnících, službách a pobočkách.
Žádosti o změny	1 řádek = 1 RfC požadavek na změnu	Žádosti o změny v závislosti na kategorii změny, pracovnících, akčním plánu, projektu, službě a pobočce.
Plánování rozpočtu	1 řádek = rozpočet na 1 službu	Náklady v závislosti na službách, akčním plánu, projektech, pobočkách, účtech a nákladových centrech.
Žádosti o nový HW a SW	1 řádek = 1 objednávka HW nebo SW	Žádosti o nový HW a SW v závislosti na druhu HW, uživatelích a pobočkách.

Tab. 4-3: Přehled žádoucí úrovně granularity faktů
Zdroj: (vlastní zpracování)

Z pohledu dimenze času se uvádí metriky ve dnech. Výjimka je spočívána v plánování rozpočtu, v němž se využívá nejvyšší granularity v řádech týdnů. Veškeré vztahy budou posléze zachyceny maticově pomocí nástroje Bus Matrix, viz příloha číslo 3.

Identifikace vícenásobných oddělených granularit pro jeden podnikový proces

Z důvodu potřeby analyzovat komplexní podnikový proces IT performance, jenž je složen z několika podprocesů a nutnosti potýkat se s jednotlivými položkami a současně i agregovanými údaji, jsme museli zvolit více tabulek faktů, v našem případě pět odrážejících počet procesů řízení IT služeb.

Identifikace typů faktové tabulky

Ve výše zmíněných pěti podprocesech můžeme nalézt dva typy faktové tabulky. V tabulce plánování zaměstnanců jsou využita na základě granularity data ve stejné úrovni podrobnosti jako v produkčních systémech, jedná se tedy o transakční typ tabulky faktů. Pro proces plánování rozpočtu je vhodněji využít vzhledem k předpokládaným metrikám a navržené granularitě sumarizační snímek od začátku rozpočtového období. V procesech zabývajících se požadavky na změnu, požadavky o přístup a objednávkami HW nebo SW se budeme zabývat workflow jednotlivých požadavků, a proto použijeme sumarizační snímky.

Ověření atomičnosti stanovené granularity

V navrhovaném systému se snažíme o co největší úroveň podrobností pro poskytnutí detailních informací. Snažení je opřeno i o současnou velikost zdrojové databáze, která je vytyčována velikostí kolem 2 gigabytů i po více jak deseti letech provozu. Množství dat na transakční úrovni a jejich budoucí růst by tak neměl výrazně ovlivnit celkovou výkonnost řešení. Vysoká úroveň detailu je předpoklad pro vytváření datových kostek nad vzniklou databází, sumarizace bude prováděna až na úrovni vzniklých kostek. Současnou granularitu můžeme přirovnat k úrovni produkčních systémů.

Identifikace předběžných kandidátů pro dimenze a fakta vycházejících z granularity

Z výše zmíněného můžeme vytvořit seznam potencionálních dimenzí - akční plány, projekty, účty, uživatelé, služby, pobočky, nákladová centra a hardware.

Současně na základě granularity určíme předběžná fakta, jako jsou dostupné a alokované pracovní hodiny pracovníků, stavy požadavků, odhad nákladů do konce roku, doba dodání HW, rychlost udělení přístupu k systému a další.

4.1.3) Identifikace dimenzí

Identifikujeme jednotlivé dimenze a jejich prvky.

Určení všech dimenzí

Na základě předchozích analýz, ověřených podrobnější studií, můžeme rozpoznat následující dimenze:

- Services,
- CostCenters,
- Accounts,
- User,
- Project,
- ActionPlan,
- Sites,
- Hardware.

Identifikace degenerovaných a sdílených dimenzí

Jak již bylo uvedeno v teoretické části, v degenerovaných dimenzích nejsou obsaženy žádné atributy. Jsou vyplněny pouze číselným identifikátorem z relačního systému, sloužící jako klíč pro seskupení atributů.

V našem případě bylo využíváno k seskupení atributů v relačním systému převážně textových nebo textově číselných identifikátorů, označujících například název služby. Výjimka je tvořena atributy požadavků na změnu (RfC) – RfcID, RfcYear a jejich spojením vzniklý RfcNo, které jsou používány pro identifikaci nově přichozího požadavku na změnu, vytvoříme pro ně jednu degenerovanou dimenzi.

Sdílené dimenze

V našem případě zatím nejsou využity sdílené dimenze v rámci celého datového skladu, ale jsou aplikovány dimenze sdílené napříč několika faktovými tabulkami v rámci návrhu dimenzionálního modelu. Konkrétně mluvíme o dimenzích, které jsou sdíleny napříč více identifikovanými fakty:

- Project,
- Sites,
- Services,
- User,
- ActionPlan,
- Accounts,
- CostCenters.

Identifikace dimensionálních atributů, granularity a hierarchií

Definujeme jednotlivé atributy dimenzí, a to v tabulkách pod čísly 4-4 až 4-12. Každá dimenze obsahuje jedinečný klíč, nastavený jako sloupec identity, a s každým vloženým záznamem do dimenze je automaticky zvýšen (inkrementován) o jedničku. Tento umělý klíč bude poskytován pro identifikaci daného záznamu ve faktové tabulce.

DIMENZE: Services			
ID: DimServices			
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	ServiceID	umělý klíč	primární klíč, auto-increment
ID služby z DB	ServiceNameID	Services.SERVICEID a RfcChange.APPID	produkční klíč
Popis služby	ServiceDescription	Services.SERVICEDESC	
Jméno podslužby	SubServiceName	ServicesSub.SERVICSUBID	produkční klíč
Popis podslužby	SubServiceDescription	ServiceSub.SERVICSUBDESC	
Příznak aktivní nebo neaktivní služby	ServiceActive	Services.ACTIVE	
Příznak aktivní nebo neaktivní podslužby	ServiceSubActive	ServiceSub.ACTIVE	
Vlastník služby	ServiceITOwner	Services.SERVICEOWNER	
Počet uživatelů službv	ServiceUserNoNumber	Services.NOUSERS	

Tab. 4-4: Návrh dimenze Services

Zdroj: (vlastní zpracování)

DIMENZE:		CostCenters	
ID:		DimCostCenters	
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	CostCentersID	umělý klíč	primární klíč, auto-increment
ID služby z DB	CostCentersNameID	CostCenters.COSTCENTERID	produkční klíč
Název služby	CostCentersName	CostCenters.COSTCENTERNAME	
Příznak aktivního nebo neaktivního nákladového centra	CostCenterActive	CostCenters.ACTIVE	

Tab. 4-5: Návrh dimenze Cost Centers

Zdroj: (vlastní zpracování)

DIMENZE:		Accounts	
ID:		DimAccounts	
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	AccountsID	umělý klíč	primární klíč, auto-increment
ID služby z DB	AccountsNameID	Accounts.AccountID	produkční klíč
Popis	AccountsDescription	Accounts.AcctDesc	
Rozpočtová kategorie	AccountsBudgetCategory	ITBudgetCategories.CATEGORY	

Tab. 4-6: Návrh dimenze Accounts

Zdroj: (vlastní zpracování)

DIMENZE:		Project	
ID:		DimProject	
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	ProjectID	umělý klíč	primární klíč, auto-increment
Název projektu	ProjectNameID	ServiceProjects.PROJECTID	produkční klíč
Příznak aktivní nebo neaktivní služby	ProjectActive	ServiceProjects.ACTIVE	

Tab. 4-7: Návrh dimenze Project

Zdroj: (vlastní zpracování)

DIMENZE:		User	
ID:		DimUser	
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	UserID	umělý klíč	primární klíč, auto-increment
ID služby z DB	UserDBName	User.DBUSER	produkční klíč
Jméno	UserFistName	USER.FIRSTNAME	
Příjmení	UserLastName	User.LASTNAME	
Název uživatele	UserWholeName	User.NAME	
Příznak aktivního nebo neaktivního uživatele	UserActive	User.ACTIVE	
Název týmu	UserTeamName	User.TEAMNAME	
Název pracovní skupiny	WorkgroupName	User.WORKGROUP	
Týdenní pracovní doba	UserWeekWorktime	User.WeekWorktime	

Tab. 4-8: Návrh dimenze User

Zdroj: (vlastní zpracování)

DIMENZE:		ActionPlan	
ID:		DimActionPlan	
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	ActionPlanID	umělý klíč	primární klíč, auto-increment
Název plánu	ActionPlanNameID	ActionPlan.ACTIONPLANID	produkční klíč
Rozpočtový rok	ActionPlanBudgetYear	ActionPlan.BUDGETYEAR	
Plánovaný začátek	ActionPlanStart	ActionPlan.STARTDATE	
Plánovaný konec	ActionPlanEnd	ActionPlan.ENDDATE	
Schválení	ActionPlanApproved	ActionPlan.APPROVED	
Popis	ActionPlanDescription	ActionPlan.DESCRPTION	

Tab. 4-9: Návrh dimenze Action Plan

Zdroj: (vlastní zpracování)

DIMENZE: Sites			
ID: DimSites			
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	SitesID	umělý klíč	primární klíč, auto-increment
Číslo pobočky	SitesSuseNumber	Sites.SUSENUMBER	produkční klíč
Jméno pobočky	SitesSiteName	Sites.SITENAME	
Obchodní jednotka	SitesBusinessUnit	BusinessUnit.ITBusinessUnit	
Oddělení	SiteBusinessDepartment	Sites.BUSINESSDEPARTMENT	
Země	SiteCountry	Sites.COUNTRY	
Region	SiteRegion	Sites.REGION	
Divize	SiteDivize	Sites.DIVISION	
Společnost	SiteCompany	Sites.COMPANY	

Tab. 4-10: Návrh dimenze Sites

Zdroj: (vlastní zpracování)

DIMENZE: Hardware			
ID: DimHardware			
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Třídící atribut	HardwareID	umělý klíč	primární klíč, auto-increment
Název hardwaru	HardwareName	ITHwTypes.HARDWARETYPE	produkční klíč
Kategorie hardwaru	HardwareCategory	ITHwCategoriesHWSTRUCTCAT	
Struktura hardwaru	HardwareStructureName	ITHwCategories.HWSTRUCTNAME	
Příznak aktivního nebo neaktivního HW	HardwareActive	ITHwTypes.HwActive	

Tab. 4-11: Návrh dimenze Hardware

Zdroj: (vlastní zpracování)

DIMENZE: RfcConCategory			
ID: DimConRfcCategory			
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	RfcCategoryID	umělý klíč	primární klíč, auto-increment
Název kategorie	RfcCategoryNameID	ITRfcConCategory.RCONCATEGORY	
Kategorie úkolu	RfcTaskCategory	ITRfcConCategory.RCONDESC	

Tab. 4-12: Návrh dimenze RfcConCategory

Zdroj: (vlastní zpracování)

Po určení všech dimenzionálních atributů stanovíme granularitu jednotlivých dimenzionálních tabulek a hierarchickou strukturu dimenzí. Určení granularity je zpracováno formou tabulky číslo 4-13: Návrh granularity dimenzionálních tabulek.

Pořadové číslo	Dimenzionální tabulka	Granularita dimenzionální tabulky
1	Project	Jsou ukládány informace o projektu, název projektu a příznak aktivity projektu.
2	Sites	Jsou ukládány informace o pobočkách společnosti.
3	Services	Jsou ukládány informace o jednotlivých službách a podslužbách.
4	User	Jsou ukládány informace o uživatelích, zahrnuje název pracovní skupiny a název pracovního týmu.
5	ActionPlan	V dimenzi jsou uchovávány informace o plánech společnosti, konkrétně název plánu, datum začátku a konce, indikátor schválení a popis daného plánu.
6	Accounts	Obsaženy informace o názvech účtů, jejich popisů a kategoriích.
7	CostCenters	Dimenze, ve kterých jsou uloženy informace o nákladových centrech, jméno centra, jeho popis a indikátor aktivity.
8	Hardware	Jsou ukládány informace o hardwaru a softwaru, jeho kategorii, typu a názvu.
9	RfcConCategory	Jsou ukládány informace o typu požadavků – název kategorie a její popis.

Tab. 4-13: Návrh granularity dimenzionální tabulek
Zdroj: (vlastní zpracování)

Identifikace hierarchií v dimenzích

V teoretické části byla formulována hierarchie jako série vztahů jednoho k mnoha a skládá se z různých úrovní, kdy každá úroveň je odpovědná atributu dimenze (Ballard, 2006).

V následující tabulce číslo 4-14 popisujeme jednotlivé dimenzionální hierarchie.

Číslo	Název	Popis Hierarchie	Typ hierarchie
1	Sites	společnost > divize > region > země > obchodní jednotka > oddělení	mírně členitá / hloubkově proměnlivá
2	Services	jméno služby > jméno podslužby	fixovaná hloubka
3	User	název pracovní skupiny > název týmu > jméno uživatele	mírně členitá / hloubkově proměnlivá
4	ActionPlan	rozpočtový rok > název plánu	fixovaná hloubka
5	Accounts	rozpočtová kategorie > název účtu	fixovaná hloubka
6	Hardware	hlavní kategorie (struktura) > kategorie > název	fixovaná hloubka

Tab. 4-14: Návrh dimenzionální hierarchie
Zdroj: (vlastní zpracování)

V rámci dimenze Project, CostCenters a RfcConCategory, nebyla objevena žádná hierarchie, z toho důvodu nejsou uvedeny výše v tabulce.

Identifikace granularity času a datumu

Granularita času a data je pokládána za důležitou pro celý návrh dimenzionálního modelu. My jsme převzali datum a čas z jiného dimenzionálního modelu jako sdílenou dimenzi napříč organizací. Dimenzi času a data nazvanou jako Calendar vidíme níže v tabulce číslo 4-15.

DIMENZE: Calendar			
ID: DimCalendar			
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
ID	[CalendarID]	sdílená dimenze	[int] NOT NULL,
Datum s časem	[Date]		[datetime] NULL,
Den	[Day]		[int] NULL,
Týden	[Week]		[int] NULL,
Měsíc	[Month]		[int] NULL,
Kvartál	[Quarter]		[int] NULL,
Rok	[Year]		[int] NULL,
Den roku	[DayOfYear]		[int] NULL,
Týden roku (CW)	[YearWeek]		[int] NULL,
Měsíc roku	[YearMonth]		[int] NULL,
Čtvrtletí roku	[YearQuarter]		[int] NULL,
Den v týdnu	[WeekDay]		[int] NULL,
Název dne	[DayName]		[varchar](15) NULL,
Název měsíce	[MonthName]		[varchar](15) NULL,
Dnů v měsíci	[DaysInMonth]		[int] NULL,
Rok, měsíc a den	[FormatYYYYMMDD]		[char](8) NULL,
Rok, měsíc a den	[FormatYYYY_MM_DD]		[char](10) NULL,
Rok a měsíc	[FormatYYYY/MM]	[char](7) NULL,	
Návrhy hierarchií:			
Rok > kvartál > měsíc > týden > den > datum s časem			
Rok > čtvrtletí roku > měsíc roku > týden roku > den v týdnu > den roku			
Název měsíce > den v týdnu > název dne			

Tab. 4-15: Datumová dimenze

Zdroj: (vlastní zpracování)

Identifikace pomalu se měnících dimenzí

Úsek bude zaměřen na identifikaci dimenzí, které se mění s ohledem na změny podmínek provozu organizace. Vzhledem k celkové „malé“ velikosti databáze a k předpokladu minimálních změn se jeví jako dostačující u většiny dimenzí přístup prepisování (SCD typ 1), odrážející změny dat v produkčním systému. Ve zdrojových tabulkách však můžeme vypořizovat i jistou modifikaci SCD typu 2, kdy je ukládán atribut indikující platnost řádku.

SCD typ 2 zachování historie, se vyskytuje v následujících případech:

- ActionPlan - ke každému plánu je ukládána doba jeho platnosti a položka Active.
- User - u neaktivních uživatelů je přepsána položka UserActive, stejný postup je využíván i v případě změny, například změnou příjmení je vytvořen nový záznam a starý záznam bude uchován s hodnotou atributu UserActive = N.
- Project - v případě změny projektu je provedeno zneplatnění záznamu ProjectActive = N a je vytvořen nový záznam.
- Services - v případě změny je nastavován atribut ServiceActive = N a stejně jako v předchozích případech je i zde vytvořen nový záznam.
- CostCentres - v případě změn je nastaven atribut CostCenterActive na hodnotu N a vytvořen nový záznam, tím je dokumentována historie.
- Hardware - předpokládá se pouze přidávání nových záznamů, pokud bude docházet ke změnám či zneplatnění záznamů, je nastavována hodnota atributu HWActive na N a vkládá se nový záznam, jímž bude zachována historie.

SCD typ 1 přepsání, využijeme u dimenzí:

- Accounts - změny se předpokládají v minimální míře, popřípadě pouze úpravou názvu účtu.
- Sites - v případě změny (například přejmenování) je považováno za vhodnější využít SCD 1 přepsání pro aktualizaci údajů napříč reporty.
- RfcConCategory - jednotlivé položky jsou standardizovány, nejsou předpokládány změny. Kdyby však byla nalezena změna, z pohledu reportingu je pokládáno za přijatelnější staré hodnoty přepsat bez uchování historie.

Identifikace rychle se měnících dimenzí (RCD)

Jedná se o každodenní změny, které nejsou zvládnuty přístupy pomalu se měnících dimenzí. Vyskytují se u atributů při nestandardizovaných vstupech uživatelů. Řešení je nalézáno v oddělení rychle se měnících atributů do samostatných dimenzí (mini dimenzí). Vytvoříme tři mini dimenze s položkami RfcDescription, RfcCategoryActivityDescription a RfcRootCause, uvedeny jsou v tabulkách pod čísly 4-16 až 1-18.

DIMENZE:	Mini_RfcActivity		
ID:	DimMini_RfcActivity		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
PopisAktivityID	RfcActivityID	umělý klíč	primární klíč, auto inkrement
Popis aktivity	RfcCategoryActivity Description	RfcConsActivity.RCOWDESCRIPTION	

Tab. 4-16: Návrh dimenze Rfc Activity

Zdroj: (vlastní zpracování)

DIMENZE:	Mini_RfcDesc		
ID:	DimMini_RfcDesc		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Popis kategoriID	RfcDescID	umělý klíč	primární klíč, auto inkrement
Popis kategorie; CR	RfcDescription	ITRfc.RFCNAME	

Tab. 4-17: Návrh dimenze Rfc Description

Zdroj: (vlastní zpracování)

DIMENZE:	Mini_RootCause		
ID:	DimMini_RootCause		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Root CauseID	RfcRootCauseID	umělý klíč	primární klíč, auto inkrement
Root Cause	RfcRootCause	ITRfc.ROOTCAUSEID	

Tab. 4-18: Návrh dimenze Rfc Root Cause

Zdroj: (vlastní zpracování)

V těchto dimenzích se předpokládá přidávání záznamů ke každému novému záznamu v dimenzionální tabulce a jejich občasné modifikace.

Identifikace případů sněhové vločky

Z teoretické části práce víme, že v relační databázi jsou uložena data ve struktuře relací normalizované do třetí normální formy. Pro dimenzionální modelování není schéma doporučováno z důvodu vlivu na pochopitelnost modelu a snížení výkonnosti při spojování rozsáhlejších tabulek.

Je známa existence dvou případů, v nichž je vyžadována nezbytnost zachovávat schéma sněhové vločky i pro dimenzionální model. V prvním, pokud dimenzionální tabulka obsahuje dva nebo více atributů rozdílné granularity. Za druhý případ pokládáme skupinu atributů stejné dimenzionální tabulky pocházející z rozdílných zdrojových systémů (Ballard, 2006).

V našem modelu se výše dva uvedené případy nevyskytují ani v jedné z dimenzí. Bylo by možno využít případu sněhové vločky, například u dimenze Sites pro položky Country, Region,

Divize, Company, u dimenze Hardware u atributů Category, Structure a Type. Ballard (2006) ale považuje za nevhodné využívat schéma sněhové vločky u právě vyjmenovaných případů.

Identifikace speciálních případů dimenzí

V průběhu tvorby multidimenzionálního datového modelu někdy mohou nastat situace, ve kterých je nezbytné zachytit specifické jevy svázané s požadavky na multidimenzionální pohled (Němec, 2014).

Níže se zaměříme na zmíněné jevy a jejich řešení ve vztahu k našemu dimenzionálnímu modelu. Po zkoumání všech případů uvedených v teoretické části práce jsme dospěli k závěru, že naším modelem jsou vykazovány znaky dimenzí s více rolemi, okrajových dimenzí a dimenzí stavových.

Dimenze s více rolemi nacházíme v případech:

User - vystupuje pro faktovou tabulku ChangeRfc v rolích ServiceMainDeveloper, RfcRootCauseBy.

Calendar – ustanoven pro faktovou tabulku AccesssDelivery v rolích DayToOrder, DayToApprove a DayToActive. Pro HardwareDelivery v rolích ToApprove, ToOrder a ToDelivery a pro faktovou tabulku ChangeRfc v rolích RfcRootCauseAt, RfcStart, RfcDelivery, RfcWorkflowStatusChang, RfcYear, DoneRelease a PlanRelease.

Okrajová dimenze vztahující se k nákladovému centru (CostCenter), zahrnuje atribut CostCenterCI, který je korelovaný numerický indikátor obsahující kromě numerických hodnot navíc tyto dva symboly „%, &“, blíže viz tabulka číslo 4-19.

DIMENZE:	Mini_CostCenters		
ID:	DimMiniCostCenteres		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	CostCentersCI_ID	umělý klíč	primární klíč, auto-increment
Identifikace CI	CostCenterCI	Budget.ACCTROWID CostRows.ACCTROWID	jedinečné hodnoty spojením všech 2 tabulek
Název CI	CostCenterCI_Name	Budget.ACCTROWNAME CostRows.ACCTROWNAME	jedinečné hodnoty spojením všech 2 tabulek
Návrhy hierarchií: Neobsahuje hierarchii			

Tab. 4-19: Návrh okrajové dimenze Cost Centers

Zdroj: (vlastní zpracování)

Následující stavové dimenze PriorityImpact, RfcScope, RfcStatusWorkflow, HardwareDeliveryStatus, AccessDeliveryStatus a AccessDeliveryReques definujeme včetně jejich atributů v následujících tabulkách pod čísly 4-20 až 4-25.

DIMENZE:	MiniPriorityImpact		
ID:	DimMiniPriorityImpact		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Generované ID	PriorityImpactID	umělý klíč	PK, auto-increment
Prioritní třída	RfcPriorityClass	ITRfc.PRIORITYCLASSID	
Vliv	RfcImpactClass	ITRfc.IMPACTCLASSID	

Tab. 4-20: Návrh stavové dimenze Priority a Impact

Zdroj: (vlastní zpracování)

DIMENZE:	Mini_RfcScope		
ID:	DimMini_RfcScope		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Rozsah ID	RfcScopeID	umělý klíč	PK, auto-increment
Rozsah	RfcScope	ITRfc.SCOPE	jedinečné hodnoty

Tab. 4-21: Návrh stavové dimenze Rfc Scope

Zdroj: (vlastní zpracování)

DIMENZE:	Mini_RfcStatusWorkflow		
ID:	DimMini_RfcStatusWorkflow		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
StatusWorkflow ID	RfcStatusWorkflowID	umělý klíč	PK, auto-increment
Workflow status	RfcStatusWorkflow	ITRfcChanges.STATUS WORKFLOW	jedinečné hodnoty

Tab. 4-22: Návrh stavové dimenze Rfc status workflow

Zdroj: (vlastní zpracování)

DIMENZE:	Mini_HardwareDeliveryStatus		
ID:	DimMini_HardwareDeliveryStatus		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
StatusID	HardwareDeliveryStatusID	umělý klíč	PK, auto-increment
Status	HardwareDeliveryStatus	ITHwOrders.STATUS	

Tab. 4-23: Návrh stavové dimenze Hardware Delivery Status

Zdroj: (vlastní zpracování)

DIMENZE:	Mini_AccessDeliveryStatus		
ID:	DimMini_AccessDeliveryStatus		
CÍL		ZDROJ	
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
StatusID	AccessDeliveryStatusID	umělý klíč	PK, auto-increment
Status	AccessDeliveryStatus	ITRfaRequest.Status	

Tab. 4-24: Návrh stavové dimenze Access Delivery Státus

Zdroj: (vlastní zpracování)

DIMENZE:	Mini_AccessDeliveryRequest		
ID:	DimMini_AccessDeliveryRequest		
CÍL	ZDROJ		
Atribut (popis)	ID	Tabulka (atribut)	Pravidla pro ETL
Identifikace	AccessDeliveryRegID	umělý klíč	PK, auto-increment
Typ požadavku	AccessDelivTypeOfRequest	ITRfaRequest.REQUESTTYPE	

Tab. 4-25: Návrh stavové dimenze Access Delivery Request

Zdroj: (vlastní zpracování)

Všechny výše uvedené typy speciálních dimenzí neobsahují žádné hierarchie.

4.1.4) Identifikace a výběr numerických faktů (metrik, ukazatelů)

Identifikujeme všechna fakta, která jsou zahrnována do dané úrovně granularity. Některá fakta jsme definovali v části věnované deklaraci podrobnosti informací o podnikovém procesu – „identifikaci předběžných kandidátů pro dimenze a fakta vycházejících z granularity“, a to:

Délka pracovních úkolů a náklady na zaměstnance, přehled dostupných a alokovaných zdrojů, přehled plánů, sledování nákladů (odhad, aktuální, odchylka), počet incidentů a jejich příčin, RfC status, stav požadavku, datum schválení, datum aktivace, počet požadavků o přístup, čas ke schválení přístupu, čas k aktivaci přístupu, schválená objednávka, datum schválení, objednání, dodání a cena.

Nyní se zaměříme na definici fakt vycházející z detailní analýzy zdrojového E/R modelu. Všechna fakta jak určená v předcházejícím kroku, tak vycházející z analýzy E/R modelu jsou uvedena v následujících čtyřech tabulkách pod čísly 4-26 až 4-29. V našem modelu se vyskytuje i případ bezfaktové tabulky faktů, konkrétně FactAccessDelivery, jejíž vypovídací část je založena v uvedení datumových údajů - objednání, dodání a aktivace, které jsou řešeny dimenzí s více rolemi a jejich sumarizací.

Fakta pro FactRFC		
Popis fakt	Fakt ID	Datový typ
Sekvence	RfcSeq	číslo
Pracovní čas	RfcWorkTime	číslo
Odhad času	RfcEstimatedTime	číslo
Proz	RfcProz	číslo
Rok	RfcYear	číslo
Identifikace Rfc	RfcID	číslo
Rok + Identifikace Rfc	RfcNo	alfanumerický
Rfc odchylka	RfcDeviation	číslo
Plánovaná práce	RfcPlannedWork	číslo

Tab. 4-26: Návrh faktové tabulky FactRfc

Zdroj: (vlastní zpracování)

Fakta pro FactWorkforceStaffPlanning		
Popis fakt	Fakt ID	Datový typ
Odhad potřebných interních pracovníků	EstIntWorkforce	číslo
Odhad potřebných nákladů za interní pracovníky	EstIntCosts	číslo
Odhad potřebných externích pracovníků	EstextWorkforce	číslo
Odhad potřebných nákladů za externí pracovníky	ExtExtCost	číslo
Dostupné pracovní hodiny	AvailableMandays	číslo
Alokované pracovní hodiny	AllocatedMandays	číslo
Odpracovaný čas pracovníků	WorkTimeHR	číslo
Odpracovaný čas pracovníků_2	TimeHours	číslo
Procentní plánovaný čas	SharePlannedWorktime	číslo
Procentní aktuální čas	ShareActualWorktime	číslo
Odhad interních MD	EstimateIntMDs	číslo
Aktuální interní MD	ActualIntMDs	číslo
Kolik procent času pracovníků je alokováno	Share	číslo
Převod pracovních hodin na dny	Days	číslo
Odchylka pracovníku času (rozdíl mezi odhadem a skutečným časem)	Deviation	číslo

Tab. 4-27: Návrh faktové tabulky FactWorkforceStaffPlanning
Zdroj: (vlastní zpracování)

Fakta pro FactHardwareDelivery		
Popis fakt	Fakt ID	Datový typ
Cena	Price	číslo

Tab. 4-28: Návrh faktové tabulky FactHardwareDelivery
Zdroj: (vlastní zpracování)

Fakta pro FactBudgetPlanning		
Popis fakt	Fakt ID	Datový typ
Odhad rozpočtu ke konci roku	EstimationEndOfYear	číslo
Aktuální rozpočet	ActualBudget	číslo
Euro rozpočet	EurBudget	číslo
Podíl rozpočtu	BudgetShare	číslo
Odhad podílu rozpočtu	EstimateShare	číslo
Aktuální podíl	ActualShare	číslo
Rozpočtové období	BudgetPeriod	číslo
Aktuální období	ActualPeriod	číslo
Rozpočtový rok	BudgetYear	číslo
Pracovní čas	TotalTime	číslo
Náklady	TotalCost	číslo
Odchylka na konci roku	DeviationEndOfYear	číslo
Podíl práce/mzdy zaměstnance na dané službě	Share	číslo

Tab. 4-29: Návrh faktové tabulky FactBudgetPlanning
Zdroj: (vlastní zpracování)

Identifikace sdílených faktů

Fakta jsou sdílena napříč několika datovými tržišti nebo sklady. V našem návrhu se nevyskytují ani jsme nevyužili žádná již sdílená fakta.

Identifikace typů faktů a agregačních pravidel

Uvnitř faktové tabulky můžeme nalézt několik rozdílných typů faktů. V této části si fakta rozdělíme do skupin definovaných v teoretické části práce. Začneme faktovou tabulkou FactWorkforceStaffPlanning, uvedenou níže pod číslem 4-30.

Fakta pro FactWorkforceStaffPlanning		
Popis fakt	Fakt ID	Typ faktů
Odhad potřebných interních pracovníků	EstIntWorkforce	<i>aditivní</i>
Odhad potřebných nákladů za interní pracovníky	EstIntCosts	<i>aditivní</i>
Odhad potřebných externích pracovníků	EstextWorkforce	<i>aditivní</i>
Odhad potřebných nákladů za externí pracovníky	ExtExtCost	<i>aditivní</i>
Dostupné pracovní hodiny	AvailableMandays	<i>aditivní</i>
Alokované pracovní hodiny	AllocatedMandays	<i>aditivní</i>
Pracovní čas pracovníků	WorkTimeHR	<i>aditivní</i>
Pracovní čas pracovníků_2	TimeHours	<i>aditivní</i>
Procentní plánovaný čas	SharePlannedWorktime	<i>neaditivní</i>
Procentní aktuální čas	ShareActualWorktime	<i>neaditivní</i>
Odhad interních MD	EstimateIntMDs	<i>aditivní</i>
Aktuální interní MD	ActualIntMDs	<i>aditivní</i>
Kolik procent času pracovníků je alokováno	Share	<i>neaditivní</i>
Převod pracovních hodin na dny	Days	<i>odvozená</i>
Odchylka pracovníku času (rozdíl mezi odhadem a skutečným časem)	Deviation	<i>odvozená</i>

Tab. 4-29: Rozdělení faktů podle typů - FactWorkforceStaffPlanning
Zdroj: (vlastní zpracování)

Faktová tabulka FactAccessDelivery je bezfaktovou tabulkou obsahující pouze cizí klíče a naopak jsou postrádána veškerá fakta. Následuje tabulka FactHardwareDelivery zobrazená pod číslem 4-31.

Fakta pro FactHardwareDelivery		
Popis fakt	Fakt ID	Typ faktů
Cena	Price	<i>aditivní</i>

4-30: Rozdělení faktů podle typů - FactHardwareDelivery
Zdroj: (vlastní zpracování)

V tabulce Fact Rfc číslo 4-32 jsou uvedena pouze fakta, jež nejsou přiřazena pomocí mini dimenze nebo dimenze více rolí. Poslední faktovou tabulku FactBudgetPlanning nalezneme pod číslem 4-33.

Fakta pro FactRFC		
Popis fakt	Fakt ID	Typ faktu
Sekvence	RfcSeq	<i>aditivní</i>
Pracovní čas	RfcWorkTime	<i>aditivní</i>
Odhad pracovního času	RfcEstimatedTime	<i>aditivní</i>
Proz	RfcProz	<i>odvozený</i>
Identifikace Rfc	RfcID	<i>neaditivní</i>
Rok + Identifikace Rfc	RfcNo	<i>neaditivní</i>
Rfc odchylka	RfcDeviation	<i>odvozený</i>
Plánovaná práce	RfcPlannedWork	<i>aditivní</i>

Tab. 4-31: Rozdělení fakt podle typů – FactRfc
Zdroj: (vlastní zpracování)

Fakta pro FactBudgetPlanning		
Popis fakt	Fakt ID	Typ faktu
Odhad rozpočtu ke konci roku	EstimationEndOfYear	<i>semi- aditivní</i>
Aktuální rozpočet	ActualBudget	<i>semi- aditivní</i>
Euro rozpočet	EurBudget	<i>aditivní</i>
Podíl rozpočtu	BudgetShare	<i>odvozená, neaditivní</i>
Odhad podílu rozpočtu	EstimateShare	<i>odvozená, neaditivní</i>
Aktuální podíl	ActualShare	<i>odvozená, neaditivní</i>
Rozpočtové období	BudgetPeriod	<i>aditivní</i>
Aktuální období	ActualPeriod	<i>aditivní</i>
Rozpočtový rok	BudgetYear	<i>aditivní</i>
Pracovní čas	TotalTime	<i>aditivní</i>
Náklady	TotalCost	<i>aditivní</i>
Odchylka na konci roku	DeviationEndOfYear	<i>odvozená, aditivní</i>
Podíl práce/mzdy zaměstnance na dané službě	Share	<i>odvozená</i>

Tab. 4-32: Rozdělení fakt podle typů - FactBudgePlanning
Zdroj: (vlastní zpracování)

Numerická fakta typu rok k datu (YTD, neboli year-to-date)

Numerické hodnoty obsahující agregace od začátku roku do aktuálního data. Jedná se o agregaci hodnoty po aktuální datum. Náš případ se týká položek aktuálního období (ActualPeriod) a aktuálního rozpočtu (ActualBudget) z faktové tabulky BudgetPlanning, které jsou agregovány už na úrovni zdrojového systému. Ostatní atributy budou agregovány až na úrovni OLAP kostek.

Událostní faktové tabulky

Využívají se pro zaznamenávání událostí, jako kliky na webové stránky (Ballard, 2006). Na základě současných požadavků se v našem modelu tento druh faktové tabulky nevyskytuje.

Návrh složeného primárního klíče

Primární klíč faktové tabulky je standardně složen z několika cizích klíčů a každý je přiřazován jedné dimenzionální tabulce. Není pokládáno za povinné mít zahrnuty všechny cizí klíče v primárním klíči. Na druhou stranu pro zajištění jedinečnosti dimenzí ve faktové tabulce je někdy vkládána potřeba zahrnout i degenerovanou dimenzi jako součást primárního klíče.

V našem modelu by neměla v zásadě nastat situace, níž je zapříčiněna nejedinečnost při kombinaci všech cizích klíčů. V případě ChangeRfc je vygenerován při opětovném požadavku nový ID požadavek.

Případ shodné žádosti v procesech AccessDelivery a HardwareDelivery by neměl nastat, byl by pouze nastolen duplicitní požadavek, jehož vznik by měl být ošetřen již na úrovni provozního systému. Blíže potřebu složeného klíče ověříme při nahrávání faktových tabulek.

Odhad velikosti a růst faktové tabulky

V teoretické části jsme uvedli dva přístupy pro odhad růstu faktové tabulky, nyní se budeme zabývat přístupem založeným na technickém pohledu, kdy budeme vycházet z knihy Rainardi (2007).

V prvním bodě sumarizujeme růsty dat za jednotlivé podprocesy. Vyházíme z 10-12leté historie fungování produkčních systémů.

FactHardwareDelivery – průměrný počet objednávek za jeden rok se pohybuje kolem 900 bez větších růstů po uplynulých 10 let. Přepokládáme stejnou tendenci růstu a přírůstek v podobně 900 řádků ročně.

FactAccessDelivery - průměrný počet požadavků je stanoven na 4500 ročně, nicméně počty požadavků jsou charakterizovány větší volatilitou a maxima je dosahováno necelých 9000 v roce 2011. Pro odhad bude počítáno spíše s růstem za poslední čtyři roky s průměrem 6500 požadavků za rok.

FactRfc - od roku 2003 je udáván průměrný počet požadavků 47 200 za rok a během posledních 4 let byl jejich průměr navýšen na 57 800. Po odstranění výkyvů je vypořizován růst a do budoucna je předpokládáno průměrně zvyšování 10% za rok.

FactBudgetPlanning – průměrný počet účtů je 2700, v průběhu historie je sledován zmíněný počet za velmi proměnlivý. Minimum je vyjadřováno počtem 402 a maximum 9500. V posledních dvou letech je viděna snaha počet zredukovat na číslo kolem 700. Můžeme odhadovat, že v nastaveném trendu bude pokračováno. Při návrhu modelu počítáme s růstem 700 řádků za rok.

FactWorkForceStaffPlanning - u tabulky můžeme vycházet z celkového počtu uživatelů, kdy podle databáze je čítáno kolem 10 000, a údaje jsou aktualizovány každý týden.

Celkem za všechny tabulky faktů by roční přírůstek měl být $900 + 6500 + 63\,580 + 700 + (10\,000 \cdot 52) = 591\,680$ řádků za rok.

Ve druhém bodě spočítáme odhad velikosti založený na růstu faktové tabulky, jako růst velikosti cizích klíčů a faktů.

Spočítání růstu faktové tabulky

Počet cizích klíčů

- WorkforceStaffPlanning: 9
- BudgetPlanning: 9
- AccessDelivery: 9
- Rfc: 24
- HardwareDelivery: 10

Počet faktů

- WorkforceStaffPlanning: 15
- Rfc: 6
- HardwareDelivery: 1
- BudgetPlanning: 13

Předpokládáme, že faktovou tabulkou je bráno 4 B pro číselný sloupec. A pro Varchar2 (10) $10 + 2\,B = 12\,B$

Celková velikost 1 řádku faktové tabulky

- WorkforceStaffPlanning $(9 + 15) \cdot 4 = 96\,B$
- BudgetPlanning $(9 + 13) \cdot 4 = 88\,B$
- AccessDelivery $(9 + 0) \cdot 4 = 36\,B$
- HardwareDelivery $(10 + 1) \cdot 4 = 44\,B$
- Rfc $(23 + 6) \cdot 4 + (1 \cdot 12) = 128\,B$

Celkový velikost přírůstku za rok

Zde využijeme údaje zjištěné v bodě jedna.

- WorkforceStaffPlanning $10\,000 \cdot 52 \cdot 96 = 49\,920\,000\text{ B} = 47,6\text{ MB}$
- BudgetPlanning $700 \cdot 88 = 61\,600\text{ B} = 60\text{ kB}$
- AccessDelivery $6500 \cdot 36 = 234\,000\text{ B} = 228,5\text{ kB}$
- HardwareDelivery $900 \cdot 44 = 39\,600\text{ B} = 38,7\text{ kB}$
- ChangeRfc $63580 \cdot 128 = 8\,138\,240\text{ B} = 7,8\text{ MB}$

Za celkový plánovaný přírůstek faktové tabulky je považováno 55,72 MB za rok.

Tímto krokem jsme ukončili proces dimenzionálního modelování, výsledek můžeme zachytit v konceptuálním modelu v rámci CASE nástroje a logickém modelu. Z důvodu velikosti modelů je můžeme oba nalézt v přílohách číslo 4 – konceptuální model a příloze číslo 5 - logický model. Struktura uvedených modelů byla konzultována se zadavatelem práce, jenž dané modely odsouhlasil a schválil, a tím byla provedena validace modelu.

4.2) Fyzický návrh

Zaměříme se na operace pro zvýšení výkonnosti databáze s pomocí agregací, indexování a rozdělení tabulek.

4.2.1) Agregační strategie

Jsou pokládány za proces počítání souhrnů dat z detailních záznamů faktové tabulky. V našem případě je poskytnuta uživatelům interaktivnější verze reportingu a nemáme zatím odezvu od uživatelů, které agregace se využívají nejčastěji. A proto bude tento krok záležitostí dalších interakcí vývoje BI řešení. Obecně se dá předpokládat, že budou využívány podobné agregace jako v původních reportech, ale dokud tento předpoklad neověříme, je preferováno agregaci zatím nerealizovat.

Druhý důvod pro neuplatnění agregační strategie je viděn ve využití datových kostek, v rámci kterých využijeme potřebná data na co nejvyšší úrovni podrobnosti a až následně bude uskutečněna jejich sumarizace.

4.2.2) Indexační strategie

V teoretické části byla formulována definice indexace, nyní ji uplatníme v praktickém příkladu. Můžeme ji rozdělit do dvou oblastí, a to na indexování dimenzí a na indexování faktových tabulek.

Indexování dimenzí

Automaticky (implicitně) jsou vytvářeny indexy na primárních klíčích dimenzí, tj. na následujících položkách: ActionPlanID, UserID, ServiceID, SitesID, ProjectID, HardwareID, CostCenterID, AccountsID, RfcConCategoryID a CalendarID.

Kromě obecných dimenzí jsou tvořeny i na primárních klíčích mini, okrajových a stavových dimenzích, konkrétně na attributech: HardwareDeliveryStatusID, CostCenterCI_ID, AccessDeliveryStatusID, AccessDeliveryRegID, RfcStatusWorkflowID, RfcScopeID, RfcImpactPriorityID, RfcRootCauseID, RfcDescID a RfcActivityID.

Explicitně je považováno za účelné vytvořit indexy na dimenzi user na attributech pracovní skupiny a týmu a na dimenzi Calendar na attributech rok, kvartál, měsíc, týden a den z důvodu většího počtu řádků a hierarchie. Ostatní tabulky jsou vyznačovány velmi malou velikostí v poměru k hierarchii a jejich indexováním by nebyla výrazně zvýšena výkonnost.

Indexování faktových tabulek

Explicitně vytvoříme zřetězené indexy na složených primárních klíčích faktových tabulek. (Primární klíče dimenzí ve faktové tabulce jsou zaznamenány jako cizí klíče, jejich spojením je dán vznik složenému primárnímu klíči faktové tabulky). Jediná výjimka je udávána faktovou tabulkou FactRfc, kde je převyšován počet cizích klíčů maximálním počtem pro indexaci, který je znázorňován 16 sloupci v rámci jednoho indexu.

4.2.3) Rozdělení tabulek (partitioning)

Rozdělování tabulek nepovažujeme v našem případě za přínosné vzhledem k předpokládané malé velikosti multidimenzionálního modelu a odhadům růstu jednotlivých faktových tabulek v řádu menším než 100 MB za rok.

Výsledek fyzického návrhu můžeme vyhledat v příloze číslo 6, v níž je zobrazen celkový fyzický model a následně jeho rozdělení do dílčích modelů na základě faktových tabulek.

4.3) Návrh a vývoj procesů a úloh datové pumpy (ETL)

Cílem je detailní definice transformačních pravidel mezi produkčními a analytickými daty. Kimball a Ross (2013) vývoj shrnují v 10 krocích.

4.3.1) Nakreslit více-úrovňový (High-level) plán

Zde navrhujeme schéma zdrojů a cílů nezávisle na konkrétní technologii a přístupu. Vzhledem k tomu, že veškeré vstupy jsou čerpány z jedné relační databáze, bude schéma velmi jednoduché. V zásadě hovoříme jen o proudu dat ze zdrojové databáze do cílové dimenzionální struktury. Výsledné schéma se nachází v příloze číslo 7.

4.3.2) Výběr ETL nástroje

Volba ETL nástroje je relativně jednoznačná v podobě Visual Studio 2010 a Microsoft SQL Management Studia, a to z následujících důvodů. Cílová databáze bude běžet nad SQL Serverem 2012 od Microsoftu a budou využívány produkty od stejného výrobce.

Za podstatnější důvod je chápáno využívání ETL nástrojů zákazníkem, který tím získá možnost budoucích úprav jednotlivých rutin dle aktuálních podnikových požadavků.

4.3.3) Vývoj výchozích strategií

S celkovým plánem požadavků (co od ETL nástroje požadujeme), můžeme vytvořit skupinu výchozích strategií pro běžné aktivity ETL systému, jedná se o následujících sedm aktivit.

Návrh extrakce dat z hlavních zdrojových systémů

Určíme výchozí metodu pro extrakci dat ze zdrojového systému. Využijeme přístupu pomocí jazyka SQL přímo do zdrojové databáze za pomoci datových pump a SQL skriptů.

Archivace extrahovaných a nezpracovaných dat

Data se nachází ve zdrojovém systému a na přání organizace budou uložena taktéž v databázi Microsoft SQL Serveru 2012. V rámci navrhovaného databázového schématu bude prováděno na nejvyšší úrovni detailu pouze základní zpracování, nebude zapotřebí uchovávat zvlášť nezpracovaná data.

Politika datové kvality pro dimenze a fakta

Kimball a Ross (2013) doporučují jako nejlepší možnost označit chybná data. Data mohou být označena pomocí auditní dimenze nebo jednoznačnou chybnou hodnotou v samotném atributu.

Stejný názor zastává i autor této diplomové práce. Za vhodný přístup k řešení chybných dat považuje označení atributů chybnou hodnotou v samotném atributu nebo využití auditních dimenzí. Samotnou implementaci provedeme v rámci návrhu datových pump.

Definice postupů zpracování změn dimenzionálních atributů

ETL systémem musí definovat, jak zvládat změny v hodnotách atributů uložených v datovém skladu včetně způsobů uchování jejich původní hodnoty. Jednotlivé způsoby zvládání změn a uchovávání historie jsou řešeny pomocí SCD, kterým jsme se věnovali v části dimenzionálního modelování v bodě identifikace pomalu se měnících dimenzí. Fyzickou implementaci SCD provedeme pomocí komponent datové pumpy v SSIS balíčku.

Zhodnocení stavu dostupnosti zdrojových a cílových systémů

Vytváření zvláštní dokumentace o dostupnosti zdrojových a cílových systémů není účelem této diplomové práce, pouze zde uvedeme, že zdrojový systém je z důvodu propojení s produkčními systémy dostupný bez odstávek. Nicméně extrakce dat z tohoto systému je vhodné provádět mimo hlavní pracovní dobu z důvodu případného negativního vlivu na výkon produkčních systémů v průběhu extrakce.

Návrh subsystému pro audit dat

Každý řádek v databázové tabulce by měl být doplněn auditními informacemi popisujícími původ dat. Tyto auditní informace budou vytvořeny pomocí popisků v podobě komentářů u jednotlivých atributů a faktů.

Organizace dočasného úložiště pro ETL

Dočasné datové úložiště pro náš navrhovaný systém nebude využito vzhledem k předpokládané velikosti růstu faktové tabulky a jednoduchým transformacím.

Tímto sedmým krokem jsme ukončili podkapitulu vývoje výchozích strategií. Následně budeme pokračovat dalšími fázemi v procesu tvorby datové pumpy.

4.3.4) Detailní analýza navrhovaného datového skladu

Obecné strategie pro běžné úlohy ETL systému byly vymezeny v předchozích částech, nyní se budeme zabývat drill-down (vnořováním) do detailních transformací potřebných pro naplnění dimenzionálních a faktových tabulek v datovém skladu.

Ověření existence čistých hierarchií (Ensure Clean Hierarchies)

Jak jsme definovali v teoretické části, jedná se o situaci, kdy žádná položka nesmí být zastoupena ve více jak jedné nadkategorii. V našem modelu jsme tyto výskyty ověřili u dimenzí s hierarchiemi a kategoriemi, konkrétně dimenze Hardware na attributech název HW, struktura a kategorie HW, dimenze Services na attributech jméno služby, jméno podslužby a dimenze User na attributech jméno uživatele, pracovní skupina a název týmu. Ověření bylo provedeno následujícím SQL příkazem uvedeným v obrázku číslo 4-2.

```
Select HardwareName, count(HardwareName) as Row_Count,  
count (DISTINCT HardwareStructureName) as Model_Count  
FROM DimHardware  
GROUP BY HardwareName  
HAVING count(DISTINCT HardwareStructureName) > 1;
```

Obr. 4-2: Ověření existence čistých hierarchií
Zdroj: (Kimball a Ross, 2013, str. 505)

Z výše uvedeného bylo zjištěno, že není vyskytován problém nečistých hierarchií v našem modelu.

Vývoj detailního schématu tabulek

V této části navrhujeme schémata tabulek s ohledem na potřebnou úroveň detailu pro drill-down. Všechny dimenzionální tabulky musí být nahrány před plněním faktové tabulky. Podrobné schéma (viz příloha číslo 9), kdy pro každou tabulku v cílovém dimenzionálním modelu byl vytvořen nahrávací úlohou pomocí SSIS balíčku.

Vývoj specifikačního dokumentu ETL

Dokument v rámci diplomové práce vytvářet nebudeme, pro konzultování se společností budou používány předchozí kroky.

4.3.5) Plnění dimenzionálních tabulek historickými daty

Začneme budováním ETL systému nejjednoduššími dimenzionálními tabulkami, jako stavové tabulky a tabulky využívající SCD typu 1. Jakmile tabulky úspěšně nahrajeme, budeme pokračovat nahráváním historických dat pro dimenze se sloupci spravovanými jako SCD typ 2.

Plnění dimenzionálních tabulek SCD typu 1

Pro historické plnění dimenzionálních tabulek využijeme SSIS balíčky vytvořené pomocí MS Visual Studia. V rámci SSIS existuje velké množství rozdílných komponent, které mohou být různě kombinovány a spojovány, výsledná aplikace je schopna vstupní data transformovat a nahrát do cílových tabulek na základě požadavků uživatele.

Před samotným nahráním byly provedeny nezbytné dimenzionální transformace v rámci SSIS balíčku, viz níže:

- Datová transformace - zhodnotili jsme, že konverzi datových typů nemáme zapotřebí, pouze jsme provedli nahrazení prázdných hodnot, a to pomocí SSIS komponenty Derived Column, která je používána pro odvození nového sloupce nebo nahrazení stávajícího. Konkrétně jsme využili v rámci této komponenty funkci „ISNULL“.
- Kombinace různých oddělených zdrojů - v našem případě využíváme pouze jeden zdroj dat bez potřeby konsolidace s dalšími zdroji. Případné zapojení více zdrojů je možno využít v dalších vývojových interakcích.
- Dekódování produkčních kódů - ve zdrojové databázi namísto produkčních kódů se převážně využívalo jedinečných textových popisů. Do našeho výsledného modelu nebyly zahrnuty produkční kódy.
- Validace vztahů mnoha k jedné a jedna k jedné - některé dimenze jsou vyznačovány jednou nebo více cestami (podmnožinami), například název HW, název typu a kategorie. U hierarchií toho typu je vyžadována podmínka jejich dokonalé čistoty. Například jednotlivé HW položky nemohou být přidružovány k několika kategoriím HW. V konkrétním případě jsme ověřily, že vztahy jsou validní a žádný níže položený prvek v hierarchii není součástí několika rodičovských kategorií.
- Přiřazení umělých klíčů dimenzím – po ověření, že dimenzionální tabulky jsou vyznačovány jedním řádkem pro každý jedinečný dimenzionální člen, můžeme přiřadit umělé klíče. Využijeme atributu auto-inkrement v rámci každé dimenze. Sloupec auto-inkrement jsme již nadefinovali v rámci vytváření dimenzionálních tabulek v SQL skriptu.

Vzhledem k podobnosti inkrementálního a historického nahrání provádíme úpravy pro historické i inkrementální nahrání, kdy odlišnost je pouze o prvotní vymazání tabulek a nahrání řádku „neznámí“. Následující operace mohou být shodné pro historické i pro inkrementální nahrání.

- Posléze využijeme komponentu Lookup z SSIS balíčku, kterou je porovnáván vstupní tok s cílovou tabulkou či pohledem na základě společných záznamů. Na základě toho porovnáme existující záznamy v tabulce, neexistující nebo shodné jen v určitých attributech. Komponenty jsou často využívány pro nahrazení hodnot shodných atributů. Pokud je nalezena shoda, je nahrazena hodnota původního atributu hodnotou atributu určené tabulky.

Plnění dimenzionálních tabulek SCD typu 2

Naším případem bude nabídnuta uživatelům možnost procházet historií v podobě, jaká jim byla umožňována prostřednictvím relační databáze. Hlubší analýza a restrukturační historie není náplní této diplomové práce. Pro plnění dimenzionálních tabulek typu SCD 2 máme v zásadě dvě možnosti. V rámci SSIS balíčku komponentu „Slowly Changing Dimension“, ve kterém naformulujeme měnící se atributy, pro odlišení platných a neplatných záznamů využijeme flagů z produkčního systému (atributů Active). Nebo přidáním další komponenty lookup, pomocí níž je provedena aktualizace vlajkového sloupce pro zneplatnění a vložení nového řádku.

Dalším krokem jsou nahrány datumové a ostatní statistické dimenze, každý datový sklad by měl zahrnovat datumovou dimenzi, obvykle na úrovni granularity jednoho řádku za den. Datumovou dimenzi přebíráme jako sdílenou entitu z jiného datového skladu včetně jejího naplnění hodnotami. Pouze jsme provedli omezení na roky v rozpětí 2002 – 2020 z toho důvodu, že data v databázi Caramba nejsou pokládána za starší než uvedené datum. Není potřeba udržovat v dimenzi kalendář starší data a maximální datum bylo vloženo s ohledem na ukládání budoucích údajů. Pokud by následně systém nebyl upraven na budoucí aktuální požadavky, byla by doplněna dimenze kalendář manuálně.

Dále bude popisována vytvořená datová pumpa pro nahrání dat do dimenzionálních tabulek, která je výsledkem vzniku předchozích kroků. Datovou pumpu jsme vytvořili v prostředí MS Visual Studio. Pro lepší představu je uvedeno celkové schéma datové pumpy v příloze číslo 8 pro historické nahrání a v příloze číslo 9 pro inkrementální nahrání, a je složena z následujících komponent:

Aktualizační procedura - je jí prováděno vymazání a nové nahrání datového úložiště vytvořeného v rámci stejné databáze jako výsledný dimenzionální model. Její odlišení od zbytku je zajištěno prefixem „stg“.

Skriptu spouštěcího proceduru uloženého v MS SQL Serveru pro vymazání cizích klíčů všech faktových tabulek - je jím umožňováno mazání dimenzionálních tabulek, které jsou uzamčeny pro mazání z důvodu vzájemných odkazů.

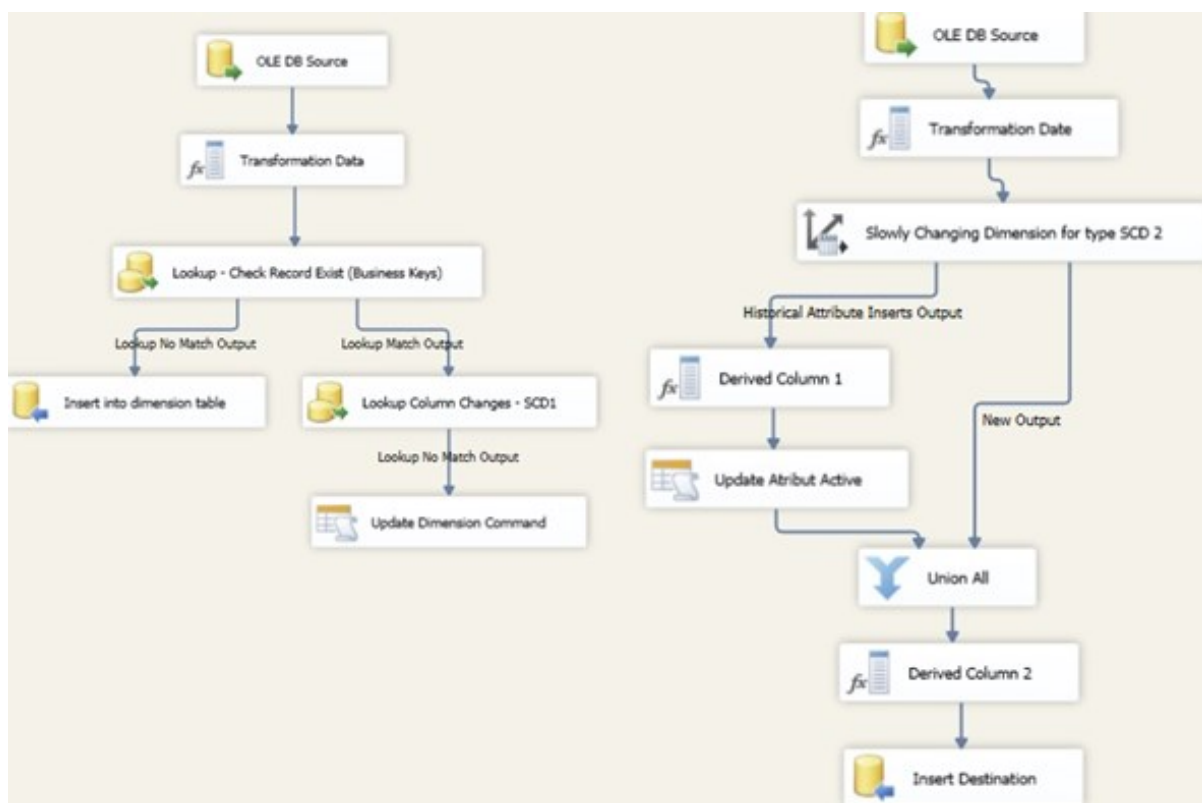
Skript (uložené procedury) pro vymazání všech dimenzí typu SCD 1 a vložení prvního výchozího řádku do dimenzí - řádek je vložen na první pozici každé dimenze, kde je užíván jako řádek „neznámí“. Řádek dále využijeme při plnění tabulky faktů, kdy údaje, kterými není shledávána shoda v klíči, budou mapovány na tento řádek.

Následuje patnáct komponent pro nahrání dat do jednotlivých dimenzí. Při jejich propojování postupujeme podle zásady, že nejdříve jsou nahrávána data do dimenzí, které neobsahují informace odvozené z jiných tabulek nebo se na ně neodkazují, pak můžeme přikročit ke zbývajícím dimenzím. V našem případě nejsou dimenze na sebe odkázány ani v nich nenajdeme odvozené informace z jiných tabulek, proto může být spuštěno nahrávání současně. Data pro nahrání jsou brána z datového úložiště s prefixem „stg“ a dle potřeby upravena.

V jednotlivých datových pumpách shledáváme značnou podobnost a v zásadě jednoduchost, proto je níže až na jeden příklad detailně neuvádíme. Před nahráním do dimenze byly provedeny opravy - odstranění prázdných řádků, konsolidace několika tabulek, vybrání duplicitních údajů, seskupení po vypuštění nepodstatných atributů a případný převod datových formátů.

Příklad jedné takové pumpy je na následujícím obrázku číslo 4-3, ve kterém vidíme OLE DB Source a OLE DB Destination sloužící jako začátek a konec datového toku. V první zdrojové komponentě je nastaven přístup ke zdrojovým tabulkám z datového úložiště a jsou určeny vstupní atributy. Je provedena transformace hodnot - nahrazení nulových hodnot, označení chybných dat a podobně. Následuje komponenta lookup, kterou jsou porovnávány hodnoty produkčních klíčů, pokud hodnoty klíče ze zdrojového systému nejsou evidovány v dimenzionální tabulce, jedná se o nový záznam, jenž vložíme do cílového úložiště. Všechny ostatní záznamy jsou přesouvány ke druhé komponentě lookup. Tou jsou porovnávány všechny atributy s dimenzionální tabulkou, pokud není nalezena shoda, byla zaznamenána změna nějakého atributu. V případě SCD typu 1 provedeme přepsání původních záznamů v tabulce pomocí příkazu update, jak je uvedeno v obrázku číslo 4-3 vlevo.

V případě SCD typu 2 uplatníme update pro zneplatnění původního záznamu (nastavíme atribut Active na N) a vložíme nový řádek. Pro SCD můžeme namísto komponenty lookup využít komponenty „slowly changing dimension“, v rámci odzkoušení byly využity obě možnosti. Ukázku SCD 2 s využitím komponenty slowly changing dimension, naleznete v obrázku číslo 4-3 vpravo, kdy následující struktura je automaticky vygenerovaná prostředím Visul Basic.



Obr. 4-3: Datová pumpa pro SCD 1 (vlevo) a SCD 2 (vpravo)
Zdroj: (vlastní zpracování)

Po nahrání všech hodnot následuje volání poslední uložené procedury datové pumpy, kterou jsou obnoveny cizí klíče všech faktových tabulek. V dalším bodě se budeme zabývat prvním plnění faktové tabulky.

4.3.6) Plnění faktových tabulek historickými daty

První nahrání faktové tabulky je významně odlišováno od následného přírůstkového nahrávání. Největší problém během historického nahrávání je spatřován ve velikosti dat, někdy až tisíckrát větší než denní přírůstek (Kimball a Ross, 2013).

- Před prvním nahráním historických dat do faktové tabulky jsme ještě jednou zrevidovali navržená fakta, pro splnění základních požadavků kladených na náš systém s pozitivním výsledkem.
- Následně jsme se zabývali otázkou auditorských statistik, kdy auditní data budeme při prvním nahrání zachytávat, ovšem kontrolu provedeme manuálně přístupem do datového skladu a zdrojového systému.
- Bylo nezbytné učinit mírné transformace faktové tabulky v oblasti záměny prázdných hodnot v klíčích a předpočtení odvozených kalkulací.

Nulová hodnota pro číselné metriky je brána jako běžná v řádcích faktové tabulky, není potřeba provádět nahrazování. Nemůžeme pouze dopustit prázdnou hodnotu ve sloupcích odkazujících na dimenze. Proto jsme vytvořili pro případ neshody v dimenzích první řádek „Unknown“ ve smyslu hodnota neznámá, kde klíčem dimenzionální tabulky bude ukazováno na tento řádek v případě nenalezení shody. Přesměrování je dosaženo při nenalezení odpovídajícího záznamu, například v dimenzi Services:

```
,DimServices_ServiceID = ISNULL(C.ServiceID, 0)
```

Bylo nezbytné provést i konverzi datových typů faktů, především jejich úpravu na vhodnou délku za desetinnou čárkou. Jednotlivá fakta jsme sjednotili na formát number (8,2).

Před nahráním tabulky faktů se musíme ujistit, že každý klíč budoucích faktových dat z produkčního systému je propojen s dimenzionální tabulkou. Propojení je realizováno pomocí umělého klíče, který jsme vytvořili nahrazením produkčních klíčů v části věnované nahrání dimenzionálních tabulek.

Při nahrání faktových tabulek FactHardwareDelivery a FactAccessDelivery, byla objevena duplicita nahrávaných klíčů, při ověření bylo zjištěno, že dochází k vytváření nových požadavků, jež se odlišovaly pouze osobou uživatele, který požadavek podal a identifikací požadavku (vygenerovaný nový produkční klíč). Byla připojena degenerovaná dimenze s produkčním klíčem požadavku, která je skládá z roku a inkrementálního čísla (vytvoření požadavku na nový HW nebo přístup), v tomto modelu je již zachycena realita.

- Ve finální fázi nahrajeme data do faktové tabulky, zde je možno řešit otázky optimálního nastavení pro nahrání, aby byly minimalizovány nároky na výkonnost. Nicméně u malých databází a přírůstků je v našem případě postrádán hlubší význam výše zmíněného kroku. Samotné nahrání řešíme pomocí SQL skriptu z důvodu ochrany přesných cest na zdrojová data, příkládána je pozměněná ukázka skriptu do přílohy číslo 10. Ostatní čtyři nahrávací skripty jsou považovány za velmi podobné, pouze jsou zaměňovány sloupce a zdrojová data, princip je ale zachován stejný, proto nejsou dále uváděna.

4.3.7) Plnění dimenzionálních tabulek přírůstkovými daty

Jednou z největších výzev spojenou s přírůstkovým ETL procesem je identifikovat nové, změněné a smazané řádky. Jakmile jsou řádky identifikovány ETL systémem, je možno aplikovat stejná pravidla transformace jako u historického nahrání.

Při navrhování SSIS balíčku pro historické nahrávání dat jsme využili komponenty lookup a SCD pro rozpoznání historických a nových záznamů, z tohoto důvodu se shoduje balíček pro přírůstkové nahrávání s SSIS balíčkem pro historické nahrávání, uvedeno v bodě 4.3.5. Jediný rozdíl nacházíme v tom, že nejsou vymazány dimenzionální tabulky příkazem TRUNCATE a není vkládán prvního defaultní řádek. Schématické znázornění nalezneme v příloze číslo 9.

4.3.8) Plnění faktové tabulky přírůstkovými daty

V tomto bodě se budeme zabývat přírůstkovým plněním tabulky faktů. Pro jeho splnění musí být faktová tabulka rozšířena o jeden sloupec zobrazující datum posledního nahrání faktové tabulky.

Extrakce dat pro faktovou tabulku a bod kontroly datové kvality - před samotnou extrakcí provedeme volání uložené procedury na SQL Serveru, již je spuštěna aktualizace mezi produkčním systémem a úložištěm v databázi MS SQL Serveru. Data zůstávají uložena z důvodu možnosti auditu a transformace bez zatěžování zdrojového systému.

Transformace faktové tabulky a přiřazení umělého klíče - nacházíme podobnost s případem historického nahrání, rozdílem je nutnost automatického zvládnutí porušování referenční integrity. Konkrétně využijeme podle teorie možnosti opravy chybného řádku na mapování, na jeden speciální řádek neznámí ‚Unknown‘ vytvořený v každé dimenzi. Ztratíme sice přehled o chybějících záznamech v tabulkách dimenzí, ale s ohledem na pravidelnou aktualizaci dimenzí je považováno řešení v rámci implementace za dostačující. V případě rozšíření systému nebo jeho pozdější modifikace je autorem práce doporučováno využít jiný způsob, a to automatické opravy chyb pomocí vytvoření řádku okrajové dimenze a vrácení umělého klíče.

Jakmile jsou extrahována data ze zdrojového systému do přechodné databáze na SQL Serveru, je potřeba identifikovat před samotným nahráním do faktové tabulky nové a změněné záznamy. Danou identifikaci je možno provést pomocí komponenty lookup. Přitom vycházíme ze znalosti, že ve zdrojovém systému se využívá sloupců auto-inkrement a provedeme v případě FactRfc, FactAccessDelivery a FactHardwareDelivery porovnání zdrojových tabulek a degenerovaných dimenzí (zahrnutých ve faktových tabulkách) obsahující produkční identifikátor, čímž jsou odhaleny nové záznamy, které jsou vloženy do faktové tabulky.

Za druhou možnost a východisko pro faktové tabulky neobsahující produkční klíč je považováno využití data ke změně řádku, které se ukládá při modifikaci, nebo k vytvoření nového řádku. V případě Oracle je datum získáváno příkazem „ORA_ROWSCN“. Vzhledem ke koncepci řešení vybíráme daný sloupec při nahrávání dat do SQL Serveru a přidáme ho do

každé z pěti hlavních tabulek, na kterých jsou postaveny faktové tabulky – jmenovitě se jedná o tabulky Rfc, HwOrder, User, Budget a RfaRequest. To provedeme upravením nahrávací procedury. Při nahrávání faktové tabulky vybereme pouze záznamy ze zdrojové databáze obsahující větší datum než datum posledního nahrání. Takto vybrané záznamy jsou dány jako nové nebo změněné.

Dalším krokem je pak oddělení nových záznamů od změněných, které bude provedeno pomocí komponenty lookup, anebo vyhledáním produkčních klíčů na shodu – pokud se nenalezne shoda, jedná se o nové záznamy, které budou vloženy do faktové tabulky. U ostatních záznamů v případě sumarizačních faktových tabulek bude proveden příkaz UPDATE.

Závěrem zaznamenáme datum posledního nahrání do faktové tabulky k nově nahaným a případně změněným záznamům.

4.3.9) Agregace dat v tabulkách a nahrání dat do OLAP databáze

Agregaci dat v tabulkách faktů pomocí vytvoření dalších agregovaných tabulek faktů řešit nebudeme. Agregace budou vytvořeny v dalším bodě číslo 4.4 při tvorbě multidimenzionálních OLAP kostek.

4.3.10) Automatizace fungování ETL systému

Ideálním ETL systémem jsou pravidelně prováděny operace nahrávání mimo pracovní dobu a bez lidské interakce, a to na základě naplánovaných úloh (jobů).

Plánování úloh (jobs) včetně jejich časové návaznosti je provedeno pomocí MS Visual Studia 2010, kde je připraveno nahrávací workflow složené z jednotlivých komponent a skriptů v jazyku SQL. Konkrétně jsou aktualizována data z produkčního systému, nahrávány dimenzionální tabulky a následně i faktové tabulky.

V dalším kroku se snažíme co nejvíce automatizovat zvládání předpověditelných výjimek a chyb, například porušení referenční integrity a obdržení nulových hodnot. Chyby zachytáváme a opravujeme v rámci ETL systému. V úvodní fázi je řešíme manuálně a po testování a zkušebním provozu automatizovaně.

Tímto krokem jsme ukončili proces návrhu a vývoje úloh datové pumpy. V následující podkapitole se budeme zabývat tvorbou datových kostek nad námi vytvořeným datovým skladem, ukázkou několika reportů a základním otestováním celého systému.

4.4) Tvorba OLAP kostek pro tabulky faktů

Základy principu byly položeny v teoretické části, nyní provedeme jejich uplatnění v modelu. Vytvoříme si celkem pět datových kostek, jednu nad každou z našich faktových tabulek. Využijeme nástroje SQL Server Business Intelligence Development Studio, konkrétně Analysis Services. Vzniklé datové kostky budou sloužit multidimenzionálnímu reportingu, zejména v softwaru Microsoft Excel.

Jako zdroje pro kostky využijeme v předchozích krocích vytvořený datový sklad ve schématu hvězdy. V rámci programu Analysis Services sestavíme nový projekt a k němu připojíme datový zdroj databázi Carambu, vytvořenou v MS SQL Serveru.

Pokračujeme připojením pěti pohledů na data, každý pohled za jednu faktovou tabulku. V následujících částech budeme popisovat jednotlivé datové kostky vzniklé z těchto pohledů.

4.4.1) OLAP kostka pro tabulku faktů FactAccessDelivery

Kostka je tvořena jednou tabulkou faktů a třemi dimenzionálními tabulkami. Konkrétně mluvíme o dimenzích Calendar, User, Services a dvou stavových dimenzích AccessDeliveryStatus a AccessDeliveryRequest. Přes uvedené dimenze jsou zobrazovány ukazatele, typu žádosti o přístup, stavu žádosti o přístup a data objednání, schválení a aktivace.

V rámci agregace kostky je možno přidat kalkulace - spočtení délky doby od objednání po schválení, doby od schválení po aktivaci a sumu těchto dvou délek. Schéma OLAP kostky je znázorněno v příloze číslo 11.

4.4.2) OLAP kostka pro tabulku faktů FactHardwareDelivery

Kostka je tvořena jednou faktovou tabulkou a čtyřmi tabulkami dimenzí, konkrétně dimenzemi User, Sites, Hardware, Calendar a jednou stavovou dimenzí Mini_HardwareDeliveryStatus, kterou jsou zobrazovány jednotlivé stavy dodávky hardwaru.

Kostkou jsou zobrazeny následující ukazatelé, datum objednání, schválení a dodání, odhadovaná cena a aktuální cena. Schéma OLAP kostky je znázorněno v příloze číslo 12.

4.4.3) OLAP kostka pro tabulku faktů FactRfc

Datová kostka je tvořena na základě jedné faktové tabulky FactRfc (Request for change) a je tvořena šesti dimezionálními tabulkami – ActionPlan, Project, User, Sites, Services a Calendar, k tomu jsou připojeny čtyři stavové dimenzionální tabulky RfcStatusWorkflow, RfcScope, ImpactPriority, RfcConCategory a tři tabulky vytvořené kvůli častým změnám a přidáváním atributů zvlášť – RfcDesc, RfcActivity, RootCause.

Struktura OLAP kostky vychází ze 13 tabulek schématu úložiště. Hlavním cílem OLAP kostky Rfc je sledovat vývoj jednotlivých požadavků a incidentů, k tomu je využíván následující seznam metrik -popis požadavku, kategorie požadavku, dopad a priorita požadavku, rozsah požadavku a stav požadavku.

V případě incidentů sledujeme jejich příčinu, datum vzniku a uživatele zodpovědného za vznik. Kromě těchto převážně textových metrik sledujeme v rámci faktové tabulky několik číselných a datumových metrik, jejich bližší vysvětlení je uváděno následujícím seznamem:

- RfcStart - datum začátku práce na požadavku,
- RfcDelivery - datum doručení hotového požadavku,
- RfcWorkflowStatusChange - datum změny stavu požadavku,
- DoneRelease - datum splnění požadavku,
- PlanRelease - plánované datum splnění požadavku,
- RfcSequence - Rfc sequence,
- RfcWorktime - aktuální odpracovaný počet hodin,
- RfcEstimatedTime - odhadovaný zbývajících počet hodin,
- RfcDeviation - rozdíl mezi odhadnutým a odpracovaným časem na splnění požadavku,
- RfcPlannedWork - plánovaný počet hodin na splnění požadavku.

Schéma propojení jednotlivých tabulek je umístěno v příloze číslo 13.

4.4.4) OLAP kostka pro tabulku faktů FactWorkforceStaffPlanning

Datová kostka s faktovou tabulkou Workforce Staff Planning je tvořena z osmi dimenzionálních tabulek – Project, Services, ActionPlan, Sites, Calendar, Cost Centers, User, Accounts a jedné okrajové dimenze Mini Cost Centers.

V rámci této kostky jsou sledovány ukazatele vážící se k jednotlivým zaměstnancům, konkrétně:

- EstIntWorkforce - odhad potřebných hodin interních pracovníků,
- EstIntCost - odhad nákladů na interní pracovníky,
- EstExtWorkforce - odhad potřebných hodin externích pracovníků,
- EstExtCost - odhad nákladů na externí pracovníky,
- AvailableMandays - dostupný počet pracovních hodnot pracovníka,
- AllocatedMandays - alokované pracovní hodiny,

- WorkTimeHR, TimeHours - odpracovaný čas pracovníků,
- SharePlannedWorktime - procentní vyjádření plánovaného času,
- ShareActualWorktime - procentní vyjádření aktuálního času,
- EstimateIntMDs - odhad dostupných interních zdrojů,
- ActualIntMDs - aktuální dostupné interní zdroje,
- Days - převod pracovních hodin na pracovní dny,
- Deviation - odchylka pracovního času (rozdíl mezi skutečným časem a odhadem).

Schéma datové kostky je umístěno v příloze číslo 14.

4.4.5) OPAP kostka pro tabulku faktů FactBudgetPlanning

Poslední datová kostka je tvořena faktovou tabulkou Budget Planning, a osmi dimenzionálními tabulkami – ActionPlan, Services, CostCenters, Sites, Calendar, Accounts, Project a User a jednou okrajovou dimenzí Mini_CostCenters.

Za účel datové kostky považujeme reporting z pohledu nákladů jak na jednotlivé pracovníky, tak i podle projektů a služeb.

V rámci samotné faktové tabulky pak jsou uchovávány následující ukazatele:

- ActualBudget - aktuální rozpočet,
- EstimationEndOfYear - odhad rozpočtu ke konci roku,
- DeviationEndOfYear - rozdíl aktuálního rozpočtu a odhadu,
- EurBudge - celkový rozpočet,
- BudgetShare - procentní vyjádření plnění rozpočtu,
- EstimateShare - procentní vyjádření odhadu plnění rozpočtu,
- ActualShare - procentní vyjádření aktuálního plnění rozpočtu,
- Budget Period - plánovaný rozpočet v rámci periody (kvartálu),
- Actual Period - aktuální rozpočet v rámci periody (kvartálu),
- BudgetYear - rozpočtový rok,
- TotalTime - celkový odpracovaný čas pracovníků,
- TotalCost - celkové náklady za odpracovaný čas pracovníků,
- Available Mandays - dostupné pracovní hodiny na konkrétní službu,
- Allocated Mandays - alokované pracovní hodiny na konkrétní službu,
- Service Share - podíl rozpočtu na dané službě.

Schéma datové kostky naleznete v příloze číslo 15.

4.4.6) Modifikace datových kostek

Po vytvoření jednotlivých datových kostek jsme provedli upravení nastavení některých dimenzí v rámci prostředí MS SAS, konkrétně:

U dimenze User jsme nastavili hierarchii: databázová zkratka uživatele > jméno a příjmení uživatele > pracovní skupina > tým.

U dimenze Services jsme upravili hierarchii na: jméno služby > jméno podslužby.

V případě Calendar jsme vytvořili dvě hierarchie: první - den > týden > měsíc > kvartál > rok. Druhá - den roku > týden roku > měsíc roku > kvartál roku > rok.

U Hardware hierarchii: název hardwaru > název sub kategorie > název kategorie.

U Sites: název pobočky > název oddělení > název divize > název společnosti > jméno země > název obchodní jednotky.

U User: databázový název uživatele > název pracovní skupiny > název pracovního týmu.

Kromě úprav v podobě hierarchií jsme vytvořili nové metriky zaměřené na počítání rozdílu v počtu dnů v případě faktových tabulek AccessDelivery a HardwareDelivery, kde počítáme rozdíl mezi dobou vzniku požadavku, schválením a aktivací. V případě Hardware Delivery od objednání přes schválení po fyzické obdržení.

Konkrétní vzniklé metriky: DayApproveToActive, DayOrderToApprove, DayOrderToActive.

Kapitola tvorby datových kostek nad námi vytvořenou databází byla ukončena. V následující části se zaměříme na demonstraci výstupů z kostek v podobě reportů a jejich následné otestování, zda odpovídají původnímu systému.

4.5) Reporting a testování

Provedeme porovnání původních generovaných statických reportů s totožným výstupem, získaným s použitím vytvořených datových kostek. V rámci původního řešení bylo možno provést vyhledávání na základě několika parametrů, konkrétně: roku, měsíce, osoby, nákladového centra, účtu, názvu služby, názvu akčního plánu, kategorii, projektu, týdne v roce (označovaného jako CW), názvu pracovní skupiny a názvu týmu. Tento výběr byl učiněn za přispění grafického rozhraní, uvedeného níže v obrázku 4-4, zobrazujícího vyhledávací parametry Caramby.

Obr. 4-4: Vyhledávací parametry Caramby
Zdroj: (pořízení jako vstupní rozhraní reportovacího systému Caramba)

Parametry mohly být voleny jen velmi omezeně, na každý report bylo voleno jen několik jistých parametrů. Reporting byl složen z předpřipravených reportů, které byly mírně modifikovány pomocí výše uvedené nabídky, kdy modifikace byla spočívána v omezení výběru na základě zadaných parametrů. Celkový počet takto předpřipravených reportů je dán číslem 44, jak můžeme vidět na obrázku číslo 4-1, uvedeném v úvodu čtvrté kapitoly.

Vytvořené řešení je založeno na trochu jiném principu, a to na možnosti uživatele vytvářet si vlastní reporty v rámci různých prostředí. Společnost preferuje prostředí v tabulkovém editoru nástroje MS Excel, jenž je využíván společností k podobným účelům. V podkapitolách budou uvedeny hlavní rozdíly Caramby a námi vyvinutého reportovacího řešení, demonstrováno pomocí MS Excel a komponenty PowerPivot. U všech reportů jsou využity fiktivní údaje nebo je provedeno nahrazení nulovými hodnotami pro zajištění ochrany dat společnosti. Kromě prezentovaného tabulkového řešení je možno vše zachytit i graficky, a to přímo v nástroji MS Excel.

4.5.1) Workforce Staff Planning

V rámci původního reportu vygenerovaného na základě zadání parametru rok do Excelu, jsme viděli následující report zobrazený v tabulce číslo 4-34.

Service	Action Plan	Start	End	MD Internal	MD External
1C Fico	2nd Level Support	1.1.2015	31.12.2015	0	0
1C Fico	Bug Fixing	1.1.2015	31.12.2015	0	0
1C Fico	Development	1.1.2015	31.12.2015	0	0
1C Fico	Interfaces	1.1.2015	31.12.2015	0	0
1C Fico	Security	1.1.2015	31.12.2015	0	0
1C Fico	Service Management	1.1.2015	31.12.2015	0	0
1C SD	Security	1.1.2015	31.12.2015	0	0
1C SD	2nd Level Support	1.1.2015	31.12.2015	0	0
1C SD	Interfaces	1.1.2015	31.12.2015	0	0
1C SD	Bug Fixing	1.1.2015	31.12.2015	0	0
1S Payroll	2nd Level Support	1.1.2015	31.12.2015	0	0

Tab. 4-33: Report - action plan

Zdroj: (pořízeno jako výstup systému Caramba, upraveno)

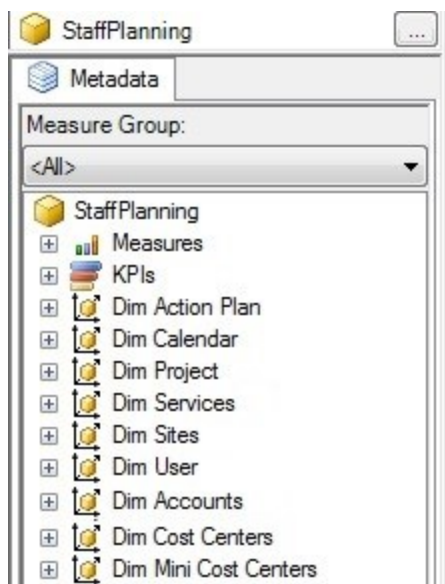
Při stejném požadavku v rámci navrhnutého řešení jsme docílili stejného výsledku zobrazení, navíc byly přidány další úhly pohledu a návaznosti, například, kterých poboček se plán týká nebo jakých uživatelů. Jednotlivé pohledy na data mohou být různě přesouvány a určovány, na základě čehož bude probíhat agregace, a to vše interaktivně přetažením položek v rámci kontingenční tabulky.

Row Labels	MD Internal	MD External
1C FiCo	0	0
2nd Level Support	0	0
2015-01-01		
2015-12-31		
Bug Fixing	0	0
Development	0	0
Interfaces	0	0
Security	0	0
Service Management	0	0
1C SD	0	0
1S Payroll	0	0
1S Trade and Stock	0	0
1st Step	0	0

Tab. 4-34: Report fact workforce

Zdroj: (vlastní zpracování)

Novým způsobem reportingu je umožňováno zobrazení stejných údajů, čímž byl splněn jeden z hlavních uživatelských požadavků, a rozšiřována možnost interaktivnosti. Základní uživatelské rozhraní pro výběr atributů v prostředí PowerPivota je uvedeno na obrázku číslo 4-5.




Obr. 4-5: Uživatelské rozhraní – workforce
Zdroj: (vlastní zpracování)

4.5.2) Budget Planning

V rámci Caramby jsou sledovány náklady v rámci pěti předpřipravených reportů, pohledy na náklady z hlediska účtů, nákladových středisek, služeb, poboček a akčních plánů. Pro demonstraci využijeme reportu z pohledu účtů a nákladových středisek, uvedeného na obrázku číslo 4-6.

IT Cost Follow Up per Account Type

Report run on: July 4, 2015 11:05 PM

STORAENSO  Page 1

Account	Cost Center	CI	Description	Actual Period	Budget Year	Budget Period	Estimate EndOfYear	Deviation EndOfYear
AD&MARK	WPS DIVFIN	%	All	0.00	0.00	0.00	0.00	0.00
	WPS DIVSWE	%	All	0.00	0.00	0.00	0.00	0.00
	Total:			0	0	0	18633,33	0
COMMUNICAT	AUT COUNTR	%	All	0.00	0.00	0.00	0.00	0.00
	AUT DIV	%	All	0.00	0.00	0.00	0.00	0.00
	CZE Countr	%	All	0.00	0.00	0.00	0.00	0.00
	EST Countr	%	All	0.00	0.00	0.00	0.00	0.00
	EST DIV	%	All	0.00	0.00	0.00	0.00	0.00
	FIN COUNTR	%	All	0.00	0.00	0.00	0.00	0.00
	FIN DIV	%	All	0.00	0.00	0.00	0.00	0.00
	RUS Countr	%	All	0.00	0.00	0.00	0.00	0.00
	SWE Countr	%	All	0.00	0.00	0.00	0.00	0.00
	Total:			0	0	0	0	0
CONSULT	AUT COUNTR	10	Not specified External Consultant	0.00	40.00	10.00	0.00	- 40.00
	CROCO	10	Vienna Aggriocultural University	0.00	2.00	0.00	0.00	- 2.00
		20	Peter Rauch	0.00	0.00	0.00	0.00	0.00

Obr. 4-6: Report Budget Planning
Zdroj: (pořizeno jako výstup systému Caramba, vlastní upravení hodnot)

Výsledkem navrženého řešení je umožňována interaktivnější práce, zobrazení všech pohledů v rámci jednoho reportu či jejich vzájemná kombinace. V rámci reportu můžeme navolit jednotlivá dělení pomocí rozklínkávání hlavních kategorií k níže položeným. Jako jednu z variant uvádíme podobný report zpracovaný v tabulce číslo 4-36.

Row Labels	Sum of Account Budget Year	Sum of Account Actual Period	Sum of Account Budget Period	Sum of Deviation End Of Year	Sum of Estimation End Of Year
AD&MARK	0	0	0	0	18633,33
WPS DIVFIN	0	0	0	0	0
Jaakko Poyry Consulting	0	0	0	0	0
Jaakko Poyry Travelling	0	0	0	0	0
WPS DIVSWE	0	0	0	0	0
COMMUNICAT	0	0	0	0	660
AUT COUNTR	0	0	0	0	0
AUT DIV	0	0	0	0	0
CZE Countr	0	0	0	0	0
EST Countr	0	0	0	0	0
EST DIV	0	330	0	330	660
FIN COUNTR	0	0	0	0	0
FIN DIV	0	0	0	0	0
RUS Countr	0	0	0	0	0
SWE Countr	0	0	0	0	0
CONSULT	123804	119652	68841,59	53271,64	172923,64
Grand Total	123804	135072	68841,59	57144,97	192216,97

Tab. 4-35: Report Budget Planning

Zdroj: (vlastní zpracování)

4.5.3) Access Delivery

Ve výchozím reportingovém systému je oblast podchycena relativně jednoduše pouhými celkovými statistikami v podobě počtu žádostí, průměrným a celkovým časem ke schválení a aktivaci služby, jak můžeme vidět na hlavičce reportu uvedené v tabulce číslo 4-37.

	A	B	C	D
1	Service	Count	SUM Time to Approve [hr]	SUM Time to Activate [hr]

Tab. 4-36: Report Access Delivery

Zdroj: (pořízeno jako výstup systému Caramba)

Pomocí nového systému je přistupováno k oblasti komplexněji, jsou sledovány jednotlivé požadavky v čase podle jejich typu, uživatele, statusu požadavku a počtu žádostí dle každé oblasti. Příklad takového reportu je zaznamenán níže v tabulce číslo 4-38, kde je uveden report na nejvyšší úrovni podrobnosti.

Service	Order Day	Approve Day	Active Day	Type Of Request	Delivery Status	Sum
Amadeus						2
	2006-05-16	2006-05-16	2006-05-17	DELETE	GRANTED	1
	2007-10-03	2007-10-03	2007-10-03	ADD	GRANTED	1
	2005-08-29	2005-08-29	2005-08-29	ADD	GRANTED	1
	2005-09-29	2005-09-30	2005-10-05	ADD	GRANTED	1
API						133
Artturi						32
Caramba						42
CZ Tool						93
Domain Admin						41

Tab. 4-37: Report Access Delivery

Zdroj: (vlastní zpracování)

4.5.4) Hardware Delivery

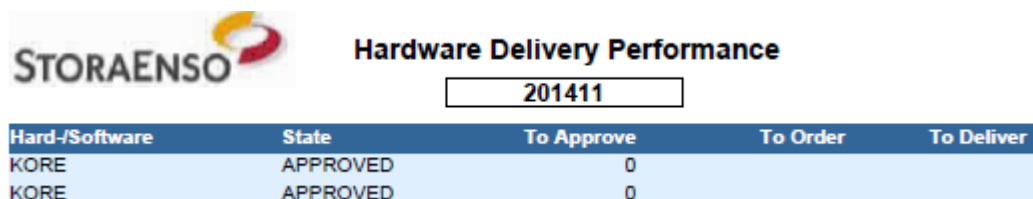
V oblasti se sleduje v původním reportingu čas od objednání hardwaru nebo softwaru po jeho schválení a doručení. Kromě těchto položek je zaměřována pozornost na název, kategorie a strukturu hardwaru, stav žádosti o hardware a měsíc vzniku žádosti. Výstupy ze systému Caramba jsou umístěny níže v obrázcích pod čísly 4-7 a 4-8.



Category	Structure	To Approve	To Order	To Delivery	Month
Cables	Other Cables	1.27	3.36	211.18	201401

Obr. 4-7: Report Hardware Delivery

Zdroj: (pořízeno jako výstup systému Caramba)



Hard-/Software	State	To Approve	To Order	To Deliver
KORE	APPROVED	0		
KORE	APPROVED	0		

Obr. 4-8: Report Hardware Delivery 2

Zdroj: (pořízeno jako výstup systému Caramba)

Nové možnosti reportingu jsou založeny v pohledech z více stran než jen v původních reportech, a to z pohledu uživatelů a poboček. Mohou být vytvářeny různé sumarizace v rámci času a zmíněných pohledů. Příklad nového reportu je uveden v tabulce číslo 4-39, kde je demonstrována možnost sumarizací.

Row Labels	Order Day	Approve Day	Delivery Day	Estimate Price	Price	Count	Deviation
DELIVERED				10399,98	9627,98	923	772,00
Cables				1280,5	1280,5	21	0,00
Other Cables				280,5	280,5	21	0,00
5m VGA cable	2010-06-21	2010-06-21	2010-08-31	10	10	1	0,00
ATEN KVM extender	2010-05-18	2010-05-18	2010-05-24	250	250	1	0,00
Avocent KVM 1x4		1	10	60	60	1	0,00
Drucker kabel paralel		1	2	27	27	1	0,00
Glass coupling (ST)		2	4	60	60	1	0,00
Communication				30467	30467	186	0,00
Computer				142520	141748	134	772,00
Computer Accessories				8020,84	8020,84	268	0,00
DENIED				4440	4440	17	0,00
REJECTED				10108	10108	22	0,00
Grand Total				24947,98	24175,98	962	772,00

Tab. 4-38: Report Hardware Delivery

Zdroj: (vlastní zpracování, hodnoty upraveny)

4.5.5) Request for change

Do oblasti byly shrnuty dvě původní oblasti, a to incident management a management změn z důvodu vzájemné provázanosti obou oblastí. Po vytvoření incidentu je pokračováno ve vytvoření požadavku na změnu. Incident změn je zachycen v následujících dvou reportech, jejich položky jsou zaznamenány v tabulkách pod čísly 4-40 a 4-41.

	A	B	C	D	E	F	G	H
1	Service	Rfc	Month	Rfc Description	State	Root Cause	Root Cause	Root Cause by

Tab. 4-39: Report Incident management

Zdroj: (pořízeno jako výstup systému Caramba)

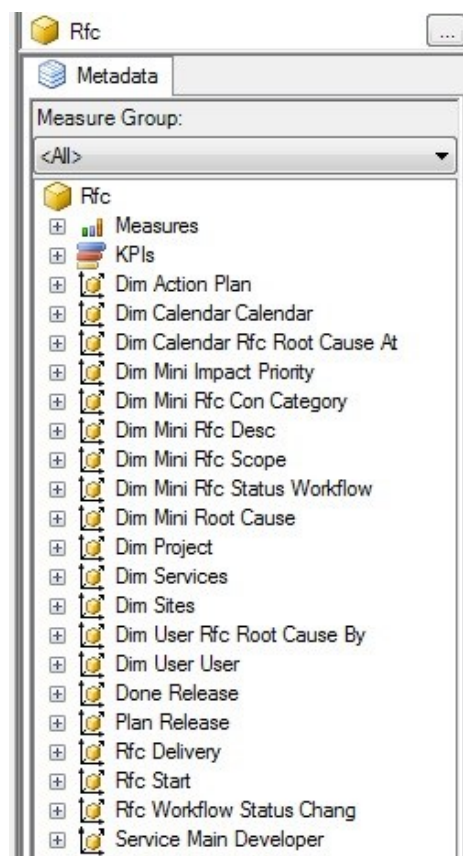
	A	B	C	D	E
1	Service	Month	Root Cause	Count	Percent

Tab. 4-40: Report Incident management 2

Zdroj: (pořízeno jako výstup systému Caramba)

V rámci našeho řešení je možno opět provádět různé sumarizace a přehledy na základě různých atributů. Oproti všem předchozím modelům je právě v tomto modelu spatřováno největší množství obdobných pohledů. Jejich nabídka je vyjadřována obrázkem číslo 4-9.

Příklad reportu je uveden v tabulce číslo 4-42, kde jsme navíc přidali oproti původním reportům pohled z hlediska jednotlivých obchodních jednotek.



Obr. 4-9: Datová kostka Rfc nabídka

Zdroj: (vlastní zpracování)

Row Labels	Service	Root Cause	Root Cause By	Root Cause At	Count
2002-04-29					
B&L	1C FiCo	Development	AALTOPI	2002-04-28	1
			AALTOSA	2002-04-28	1
			ADAMERU	2002-04-29	1
			ABROMDA	2002-04-29	1
			CIOO	2002-04-29	1
		Never worked			10
		Not found			5
		Release			3
	1S Payroll				5
	1S Trade and Stock				20
	1st Step				3
	AA-Austria				5
	AA-Corbeham				10
FP					50
PUUM					50
WS FIN					30
2002-04-30					330

Tab. 4-41: Report Incident management

Zdroj: (vlastní zpracování, hodnoty upraveny)

Tímto je ukončena část diplomové práce věnující se porovnávání řešení. V rámci jednotlivých reportů před využitím fiktivních hodnot jsme ověřili, že jednotlivé hodnoty obou reportingových prostředí jsou sobě navzájem odpovídající. Kromě ověření na výše uvedených reportech byla tato skutečnost přezkoumána i na ostatních reportech z prostředí Caramby.

Nyní se budeme zabývat testováním z pohledu splnění uživatelských požadavků, definovaných v rámci prvního bodu dimenzionálního modelování v podkapitole číslo 4-1-1. K tomu využijeme shodné tabulky požadavků, jaké jsme dříve definovali, pouze provedeme jejich upravení. Odstraníme nepotřebné sloupce a nahradíme je sloupci obsahujícími informace o splnění daných požadavků, v jakém z pěti modelů jsou požadavky splněny a pomocí jakých metrik.

Pro lepší orientaci v tabulce číslo 4-43 využijeme číslování pěti datových kostek/faktových tabulek od jedničky do pětky, viz následující seznam:

- Model 1 – AccessDelivery,
- Model 2 – HardwareDelivery,
- Model 3 – Budget Planning,
- Model 4 – Workforce Staff Planning,
- Model 5 – Request for Change.

Č.P.	Požadavek	Měřitelné hodnoty (ukazatele)	Požadavek splněn	M.	Metrikou
Q1	sledování incidentů a jejich příčin	počet incidentů a jejich příčin	ANO	5	RootCauseAt, RootCauseBy, RootCauseID, součet incidentů
Q2	sledování RfH statistik	počet objednaného HW	ANO	2	součet objednávek HW
Q3	sledování RfA statistik	počet požadavků o přístup	ANO	1	součet požadavků
Q4	přehled plánů	přehled plánů	ANO	4	ActionPlan (dimenze)
Q5	alokace zdrojů	přehled dostupných a alokovaných zdrojů	ANO	4	AvailableMandays, AllocatedMandays
Q6	výkonost dodávek HW	počet objednaných a schválených HW	ANO	2	součet s vyplněným datem objednání a datem schválení
Q7	sledování přístupové doby služeb	čas ke schválení přístupu, čas k aktivaci přístupu	ANO	1	DayToOrder, DayToAccept, DayToActive
Q8	sledování plnění rozpočtu na základě účtů, nákladových center, služeb	sledování nákladů (odhad, aktuální, odchylka)	ANO	3	ActualBudget, EstimationEndOfYear, DeviationEndOfYear
Q9	pracovní doba zaměstnanců v závislosti na obchodních jednotkách, službách, plánech	délka pracovních úkolů a náklady na zaměstnance	ANO	4	EstIntWorkforce, EstextWorkforce, EstIntCost, EstExtCost
Q10	stav vyřizování požadavků	Rfc status	ANO	5	RfcWorkflow, StatusChange, RfcStatusWorkflow, RfcStart, RfcDelivery

Tab. 4-42: Analýza plnění uživatelských požadavků
Zdroj: (vlastní zpracování)

Z výše uvedené tabulky 4-34 vyplývá, že jsme splnili uživatelské požadavky kladené na výsledný systém a testování splnění požadavků můžeme pokládat za úspěšné.

V další fázi bylo přikročeno k představení řešení společnosti, kdy první ohlasy jsou vnímány pozitivně a řešením jsou splněny její představy. Je pokračováno obdobím testování uživateli a případné doladování a zkušební provoz.

Poslední kapitolu diplomové práce můžeme považovat za ukončenou. V ní jsme se věnovali reportingu a testování námi navrženého řešení. Byly provedeny testy na všech pěti datových kostkách s pozitivním výsledkem a otestováno splnění všech uživatelských požadavků.

5) Závěr

Cílem diplomové práce bylo navrhnout vhodné řešení pro zlepšení efektivity řízení IT služeb ve výrobním podniku. Cíl byl řešen podrobnou analýzou procesů a služeb společnosti, vybráním optimálního procesu, kterým by byl naplněn účel práce a přinesl společnosti přidanou hodnotu. Po výběru vhodného procesu byla provedena jeho detailní analýza, určeny nedostatky současného řešení, kdy za hlavní nedostatek byla považována malá možnost interaktivního reportingu. Řešením tohoto nedostatku bylo vytvoření business intelligence řešení, kterým byla zvýšena interaktivita a přineseny nové možnosti pro reporting.

Pro pokrytí řešení bylo využito Kimballova životního cyklu business intelligence aplikace, který byl složen z několika oblastí. V úvodu procesu byla řešena analýza uživatelských požadavků metodou interview s uživatelem systému a pomocí hloubkového rozboru současného řešení.

V druhé části diplomové práce jsme se zabývali oblastí dimenzionálního modelování, která byla složena ze čtyř bodů. V prvním bodě jsme rozšířili původní analýzu vybraného procesu, identifikovali sdílené entity a metriky, zdroj dat včetně využití zpětného inženýrství pro identifikaci zdrojů jednotlivých datových položek ze starého řešení a provedli sumarizaci a konkretizaci uživatelských požadavků. Ve druhém bodě jsme deklarovali, při jaké podrobnosti informací budeme daný proces modelovat, to znamená, určili jsme granularitu faktové tabulky, její typ a na základě určené granularity identifikovali předběžné kandidáty pro dimenze a fakta. Ve třetím bodě jsme identifikovali dimenze skládající se z určení degenerovaných a sdílených dimenzí, atributů, granularity a hierarchií dimenzí. Stanovili jsme úroveň podrobnosti času a data, identifikovali pomalu a rychle se měnící dimenze, případy sněhové vločky a speciální případy dimenzí. Ve čtvrtém bodě jsme se věnovali identifikaci numerických faktů, kde jsme se zabývali sdílenými fakty, určením typů fakt a agregačními pravidly, událostními faktovými tabulkami, návrhem složeného klíče a odhadem velikosti růstu faktové tabulky.

Ve třetí části jsme řešili fyzický návrh, zabývali jsme se návrhem agregační strategie, indexační strategie a rozdělením tabulek.

Čtvrtou část jsme se věnovali návrhu a vývoji procesů a úloh datové pumpy. Zmíněný proces je složen z deseti kroků a za jejich výsledek považujeme navržené jednotlivých datových pump pro dimenzionální a faktové tabulky. Závěr byl zaměřen na automatizaci procesu fungování datové pumpy.

Po dokončení Kimballova životní cyklu jsme navrhli datové kostky, celkem bylo navrženo pět datových kostek, jedna pro každou faktovou tabulku.

Závěrečná část diplomové práce byla věnována porovnávání původních reportů s možnostmi, které přineslo nové řešení, testování správné funkčnosti, ověření splnění požadavků a závěrečná zhodnocení.

Přínos své práce autor shledává především v podrobné analýze procesů společnosti, zjištění vhodného procesu pro zlepšení reportingu, návrhu a realizaci řešení business intelligence. Výsledné navrhnuté řešení významně zlepšuje vypovídající schopnost a interaktivitu reportingu.

Práce naplnila svůj účel vytvořením řešení, kterým je zefektivňováno řízení IT služeb, zvyšována vypovídající hodnota dat a je zaváděna vyšší interaktivnost pro reporting. Všechny stanovené cíle byly splněny. Autor diplomové práce věří, že výsledné řešení bude společností plně využíváno a přispěje tak k efektivnější kontrole, plánování a rozhodování.

Námětem pro další navazující práci může být zaměření se na datovou kvalitu, rozšíření datových zdrojů a zpracování automatizovaných grafických reportů založených na vytvořených datových kostkách.

Zdroje

Tištěné

KIMBALL, Ralph a Margy ROSS. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3. vyd. Indianapolis: Wiley, 2013. ISBN 978-1-118-53080-1.

BALLARD, Chuck, Daniel FARRELL, Amit GUPTA, Carlos MAZUELA a Stanislav VOHNIK. *Dimensional modeling: In a business intelligence environment*. IBM International Technical Support Organization, 2006. ISBN 07-384-9644-8.

GOLFARELLI, Matteo a Stefano RIZZI. *Data warehouse design: Modern principles and methodologies*. New York: McGraw-Hill, 2009. ISBN 978-0-07-161039-1.

BALLARD, Chuck. *Data modeling techniques for data warehousing*. United States: IBM, 1998. ISBN 978-073-8402-451.

NĚMEC, Radek. *Principy projektování a implementace systémů Business Intelligence*, SOET, vol. 11. Ostrava: VŠB-TU Ostrava, 2014. ISBN 978-80-248-3452-8.

KIMBALL, Ralph a Joe CASERTA. *The data warehouse ETL toolkit: Practical techniques for extracting, cleaning, conforming, and delivering data*. Indianapolis: Wiley, 2004. ISBN 07-645-6757-8.

INMON, William H. *Building the data warehouse*. 4. vyd. Indianapolis: Wiley, 2005. ISBN 07-645-9944-5.

NOVOTNÝ, Ota, Jan POUR, Miloš MARYŠKA a Jasef BASL. *Řízení výkonnosti podnikové informatiky*. Praha: Professional Publishing, 2010. ISBN 978-80-7431-040-9.

NOVOTNÝ, Ota. *Business intelligence: Jak využít bohatství ve vašich datech*. Praha: Grada, 2005. ISBN 80-247-1094-3.

POUR, Jan, Miloš MARYŠKA a Ota NOVOTNÝ. *Business intelligence v podnikové praxi*. Praha: Professional Publishing, 2012. ISBN 978-80-7431-065-2.

TVRDÍKOVÁ, M. *Zavádění a inovace informačních systémů ve firmách*. Praha: Grada, 2000. ISBN 80-716-9703-6.

VERCELLIS, Carlo. *Business intelligence: Data mining and optimization for decision making*. Chichester, UK: Wiley, 2009. ISBN 04-705-1139-7.

RAINARDI, Vincent. *Building a data warehouse with examples in SQL Server*. New York: Berkeley CA. ISBN 15-905-9931-4.

KALUŽA, Jindřich a Ludmila KALUŽOVÁ. *Modelování dat v informačních systémech*. Praha: Ekopress, 2012. ISBN 978-80-86929-81-1.

Elektronické

JETHWA, Niles. How to clean Oracle GL Hierarchies. *IT Toolbox* [online]. 2007 [cit. 2015-06-27]. Dostupné z: <http://it.toolbox.com/blogs/bi-applications/how-to-clean-oracle-gl-hierarchies-13889#commentAnchor>

KALOUSKOVÁ, E.; POLÁKOVÁ, J.; Data, informace, znalosti – rozdíly, podobnosti. [online]. [cit. 2011-03-2]. Dostupné z: <http://www.knowledgemanagement.ic.cz/informaceznalosti.doc>

MOLNÁR, Zdeněk. Manažerské informační systémy: Úvod a teoretická východiska. Slideplayer [online]. 2009 [cit. 2015-04-28]. Dostupné z: <http://slideplayer.cz/slide/2343806/>

Oracle Olap. Gerardnico [online]. 2009 [cit. 2015-04-28]. Dostupné z: http://gerardnico.com/wiki/database/oracle/oracle_olap

Oracle. Defining Dimensions. Oracle9i OLAP Developer's Guide to the OLAP DML [online]. 2002 [cit. 2015-05-10]. Dostupné z: https://docs.oracle.com/html/A95298_01/define4.htm

STORA ENSO. 2013. *The largest Stora Enso IT centre* [online]. [cit. 2015-05-14]. Dostupné z: <http://extra.storaenso.com/ostrava/>

STORA ENSO. 2015. *Stora Enso* [online]. [cit. 2015-05-14]. Dostupné z: <http://www.storaenso.com/>

ŠLAPÁK, Ondřej. Data, informace, znalosti [online]. 2003. [cit. 2015-05-14]. ISSN 1211-0442. Dostupné z: <http://nb.vse.cz/kfil/elogos/miscellany/slapa103.pdf>

Business dictionary. What is business process? Definition and meaning [online]. 2015 [cit. 2015-04-28]. Dostupné z: <http://www.businessdictionary.com/definition/business-process.html>

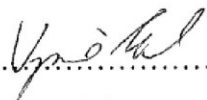
Seznam Zkratek

BI	Business Intelligence
CPM	Corporate Performance Management
DMA	Data Marts (datová tržiště)
DSA	Data Staging Area (dočasná úložiště dat)
DSS	Decision Support Systems (systém pro podporu rozhodování)
DWH / DW	Data Warehouses (datový sklad)
EAI	Enterprise Application Integration (aplikační integrace)
EIS	Executive Information System (systémy pro podporu vrcholového řízení)
ES	Expertní systémy
ETL	Extract, transformation, Load
ETL	Extraction Transformation Loading (transformační nástroje – extrakce, transformace, nahrání)
FK	Foreign Key (cizí klíč)
HOLAP	Hybrid online analytical processing
HW	Hardware
ID	Identification
KPI	Key performance Indicator (klíčové indikátory výkonnosti)
KRI	Key Result Indicator (klíčové indikátory výsledků)
MIS	Management Information System (manažerské informační systémy)
MOLAP	Multidimensional online analytical processing
ODS	Operational Data Store (operativní úložiště dat)
OLAP	On-Line Analytical Processing
OLTP	On-Line Transaction Processing
PI	Performance Identifiers (ostatní indikátory výkonosti)
PK	Primary Key (primární klíč)
ROLAP	Relational online analytical processing
SCD	Slowly changing dimension (pomalou se měnící dimenze)
SW	Software
ISO	International Organization for Standardization
ERP	Enterprise Resource Planning (podnikové informační systémy)
SCM	Supply Chain Management (systémy řízení dodavatelského řetězce)
CRM	Customer relationship management (systémy řízení vztahů se zákazníky)
E/R model	Entity-relationship model
SQL	Structured Query Language

Prohlašuji, že

- jsem byl seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že bibliografické údaje o diplomové práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
-

V Ostravě dne 14.7.2015

.....
Radek Vymětal

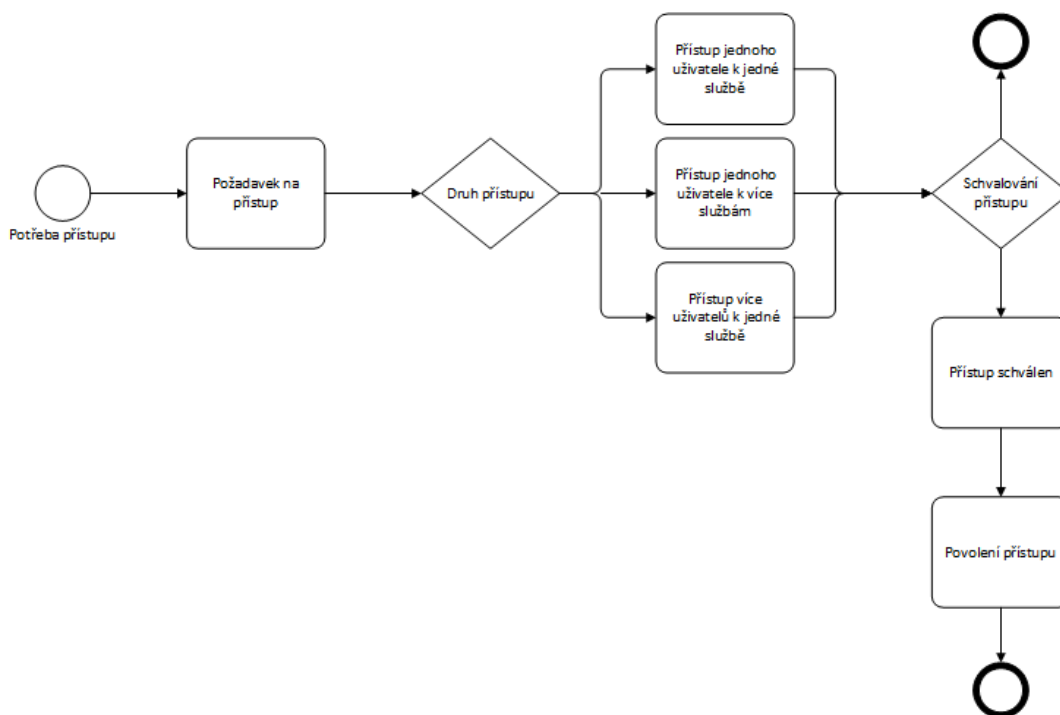
Seznam příloh

PŘÍLOHA 1.....	BPMN PROCESNÍ MODEL Y
PŘÍLOHA 2.....	ŠABLONA PROCEDURY PRO NAHRÁVÁNÍ DAT
PŘÍLOHA 3.....	BUS MATRIX
PŘÍLOHA 4.....	CELKOVÝ KONCEPTUÁLNÍ MODEL
PŘÍLOHA 5.....	CELKOVÝ LOGICKÝ MODEL
PŘÍLOHA 6.....	CELKOVÝ FYZICKÝ DATOVÝ MODEL
PŘÍLOHA 7.....	VYSOKO-ÚROVŇOVÝ PLÁN (HIGH-LEVEL PLAN)
PŘÍLOHA 8.....	DATOVÁ PUMPA DIMENZIONÁLNÍHO MODELU – HISTORICKÉ NAHRÁNÍ
PŘÍLOHA 9.....	DATOVÁ PUMPA DIMENZIONÁLNÍHO MODELU - INKREMENTÁLNÍ NAHRÁNÍ
PŘÍLOHA 10.....	SQL SKRIPT PRO NAHRÁNÍ FAKTOVÉ TABULKY RFC
PŘÍLOHA 11.....	OLAP KOSTKA PRO FAKTOVOU TABULKU FACTACCESSDELIVERY
PŘÍLOHA 12.....	OLAP KOSTKA PRO FAKTOVOU TABULKU FACTHARDWAREDELIVERY
PŘÍLOHA 13.....	OLAP KOSTKA PRO FAKTOVOU TABULKU FACTRFC
PŘÍLOHA 14.....	OLAP KOSTKA PRO FAKTOVOU TABULKU FACTWORKFORCESTAFFPLANNING
PŘÍLOHA 15.....	OLAP KOSTKA PRO FAKTOVOU TABULKU FACTBUDGETPLANNING

Příloha 1: BPMN procesní modely

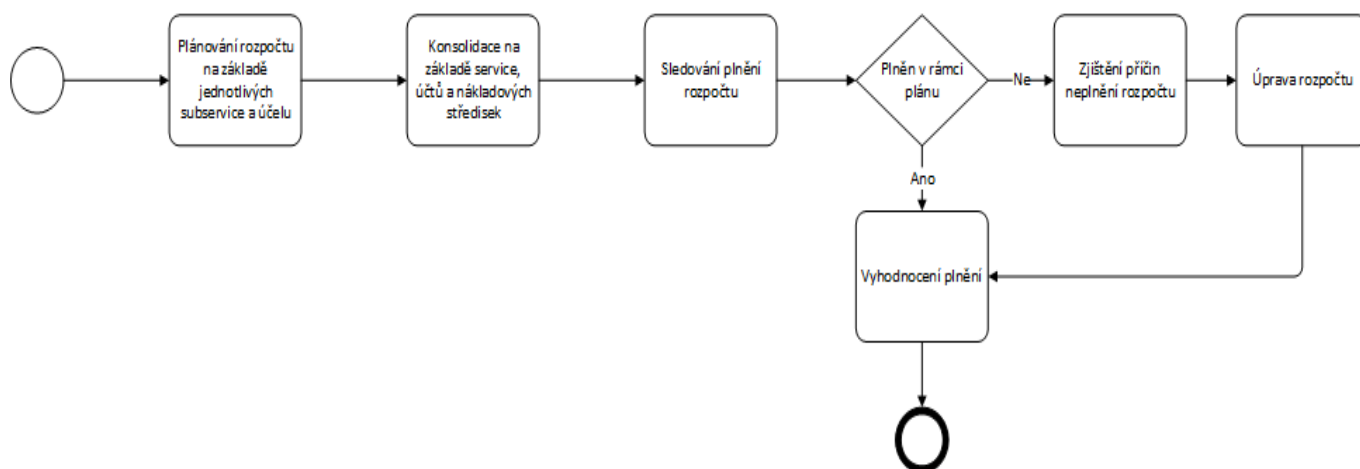
a) Proces udělení přístupu

Proces udělení přístupu

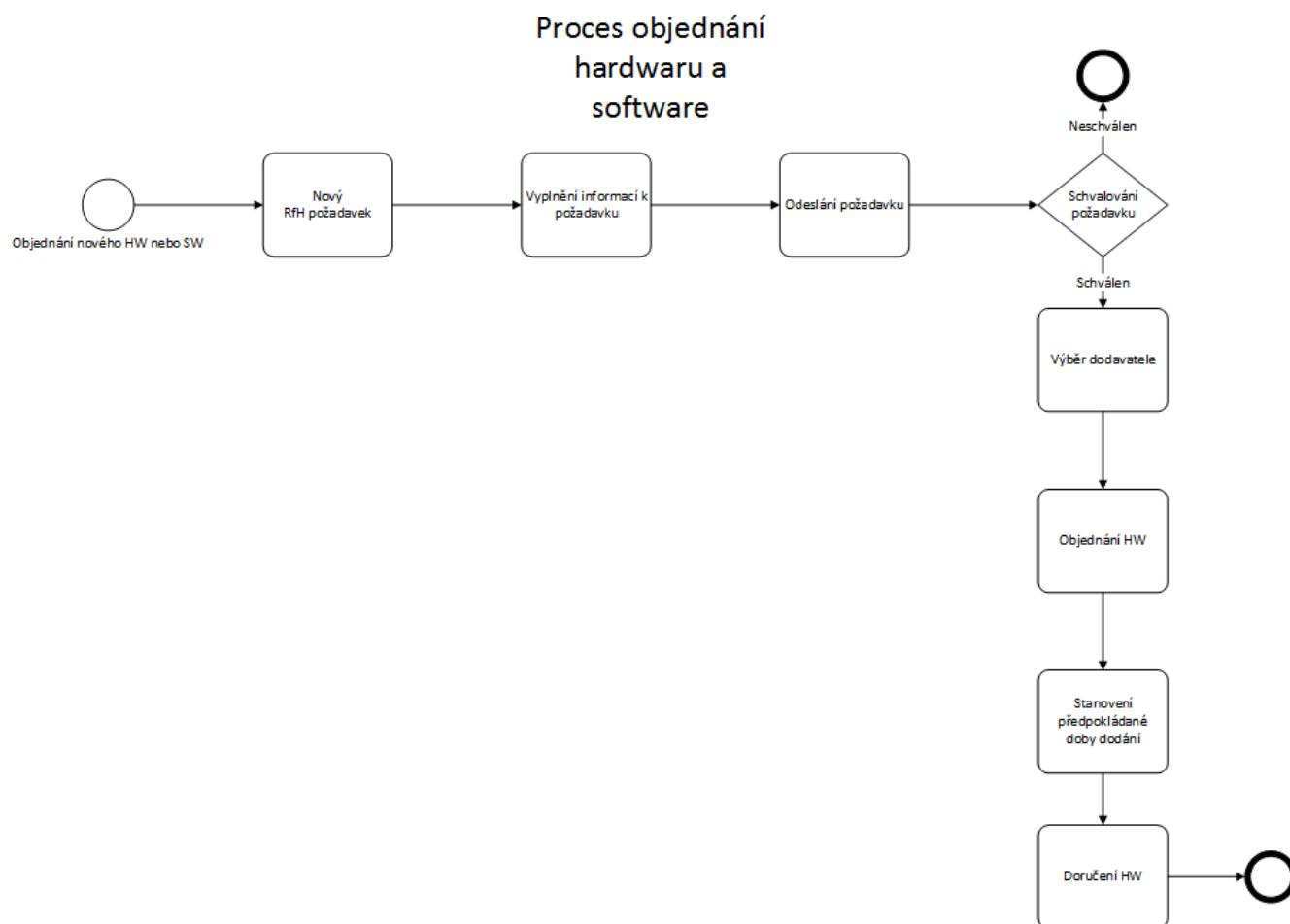


b) Proces plánování rozpočtu

Proces plánování rozpočtu

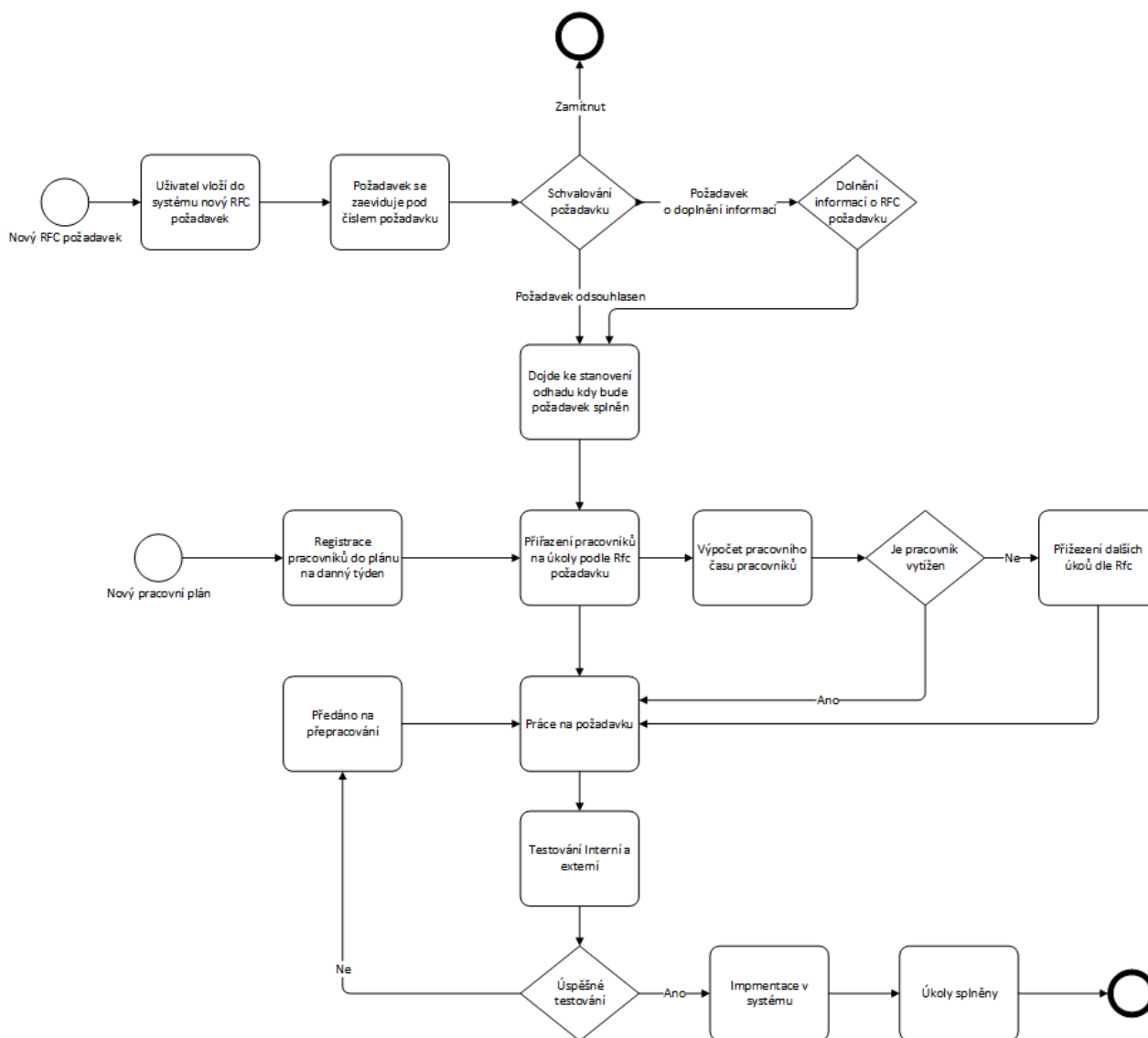


c) Proces objednávání hardwaru a softwaru



d) Proces managementu změn a plánování úkolů zaměstnanců

Management změn - RFC a plánování úkolů zaměstnanců

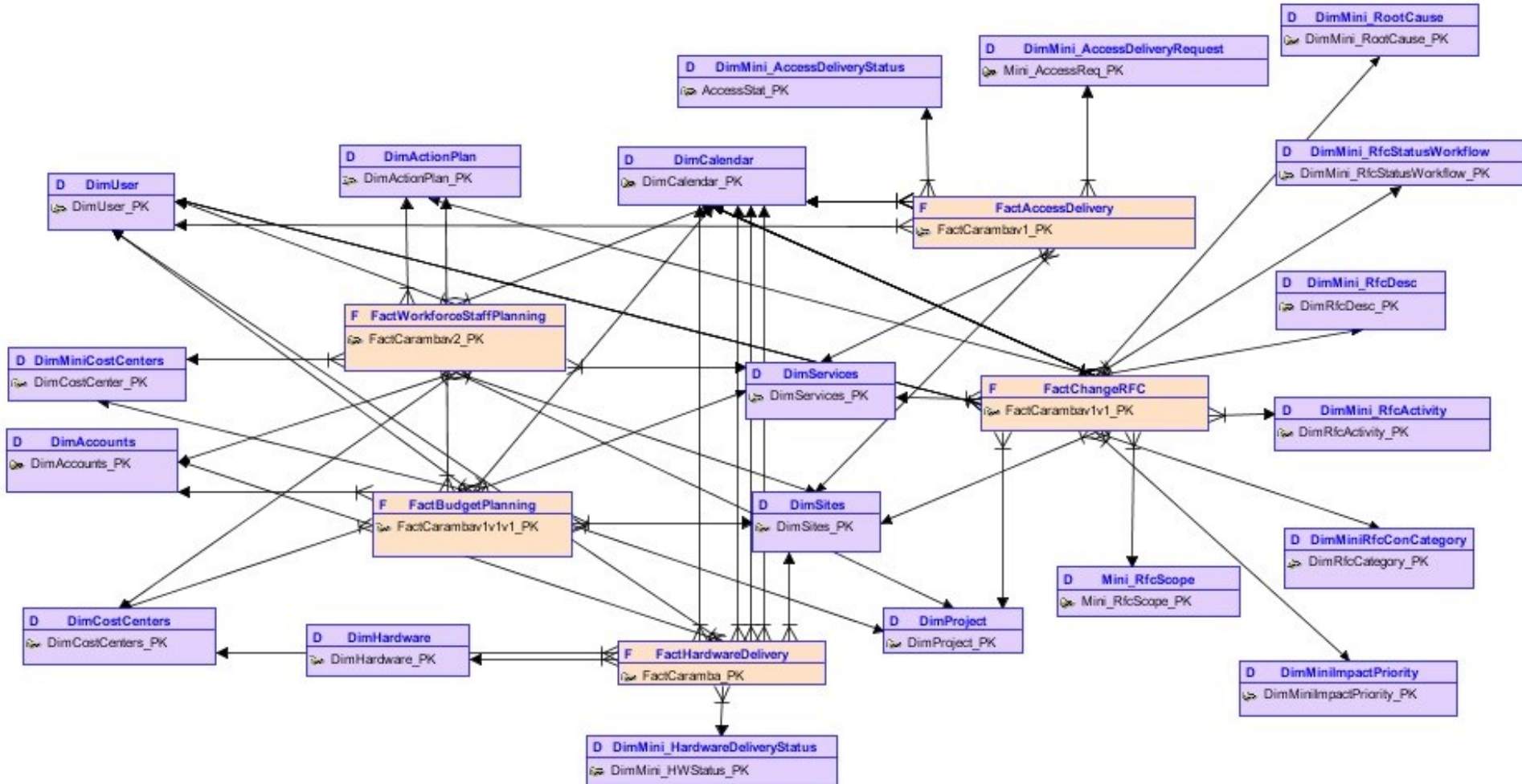


Příloha 2: Šablona procedury pro nahrávání dat

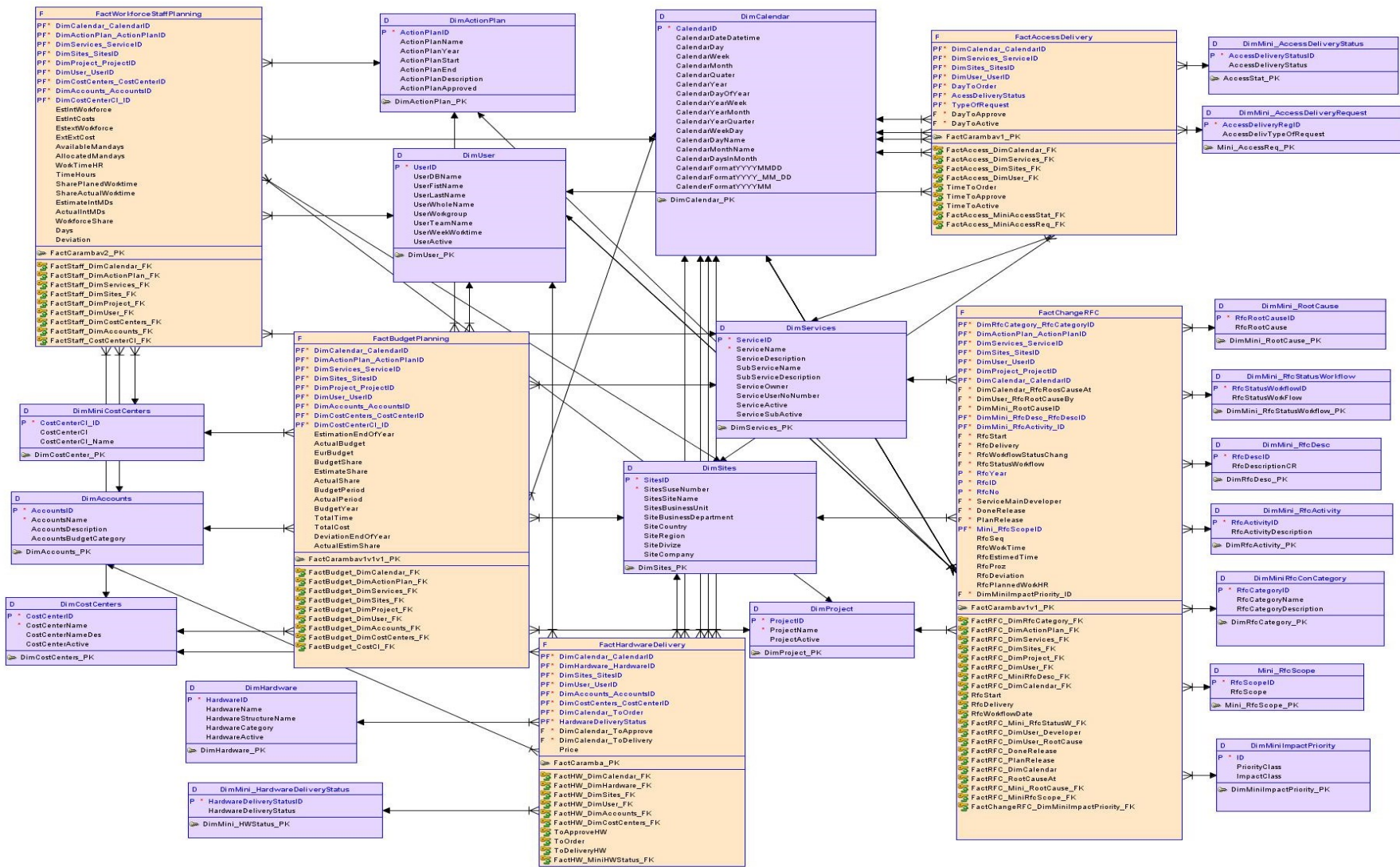
```
1 DECLARE @table VARCHAR(100)
2 SELECT @table = 'NameOfTable'
3
4 DECLARE @table_id INT
5 SELECT @table_id = object_id |
6 FROM sys.tables
7 WHERE schema_id = SCHEMA_ID('stg')
8 AND name = @table
9
10 DECLARE @SQL VARCHAR(MAX)
11 SELECT @SQL = ''
12 SELECT @SQL = @SQL + 'SET ANSI_NULLS ON' + CHAR(13)
13 SELECT @SQL = @SQL + 'GO' + CHAR(13)
14 SELECT @SQL = @SQL + 'SET QUOTED_IDENTIFIER ON' + CHAR(13)
15 SELECT @SQL = @SQL + 'GO' + CHAR(13)
16 SELECT @SQL = @SQL + CHAR(13)
17
18 SELECT @SQL = @SQL + 'CREATE PROCEDURE dm.usp_DM_Dimension' + @table + '_Load' + CHAR(13)
19 SELECT @SQL = @SQL + 'AS' + CHAR(13)
20 SELECT @SQL = @SQL + 'BEGIN' + CHAR(13)
21 SELECT @SQL = @SQL + CHAR(13)
22 SELECT @SQL = @SQL + '    SET NOCOUNT ON;' + CHAR(13)
23 SELECT @SQL = @SQL + CHAR(13)
24 SELECT @SQL = @SQL + '    TRUNCATE TABLE dm.' + @table + ';' + CHAR(13)
25 SELECT @SQL = @SQL + CHAR(13)
26 SELECT @SQL = @SQL + '    INSERT INTO dm.' + @table + '('
27
28 SELECT @SQL = @SQL + ' ' + name + ','
29 FROM sys.columns
30 WHERE object_id = @table_id
31
32 SELECT @SQL = @SQL + ')' + CHAR(13)
33 SELECT @SQL = @SQL + '    SELECT' + CHAR(13)
34 SELECT @SQL = @SQL + '        ' + name + ',' + CHAR(13)
35 FROM sys.columns
36 WHERE object_id = @table_id
37 SELECT @SQL = @SQL + '    FROM stg.' + @table + CHAR(13)
38 SELECT @SQL = @SQL + 'END'
39 print @SQL
40 EXEC @SQL
```

Proces	Tabulky faktů	Granularita	Fakta	Kalendář	Akční plán	Services	Uživatelé	Pobočky (místa)	Projekty	Rfc kategorie	Hardware	Účty	Nákladové centrum
IT performance	Plánování zaměstnanců	1 řádek = pracovní plán zaměstnance	Délka pracovních úkolů a náklady na zaměstnance, přehled dostupných a alokovaných zdrojů, přehled plánů	X	X	X	X	X	X			X	X
	Plánování rozpočtu	1 řádek = rozpočet na 1 službu	Sledování nákladů (odhad, aktuální, odchylka)	X	X	X	X	X	X			X	X
	Požadavek na změnu	1 řádek = 1 Rfc požadavek na změnu	Datum vzniku incidentu, zodpovědná osoba, popis incidentu, popis požadavku, popis aktivity spojené s požadavkem, workflow status požadavku, čas práce na požadavku, identifikace požadavku, plán implemetance	X	X	X	X	X	X	X			
	Požadavek přístupu	1 řádek = 1 požadavek na přístup	Datum objednání, stav požadavku, typ požadavku, datum schválení, datum aktivace,	X		X	X	X					
	Objednávka HW nebo SW	1 řádek = 1 objednávka HW nebo SW	Stav dodání, datum schválení, objednání, dodání, cena	X		X		X			X	X	X

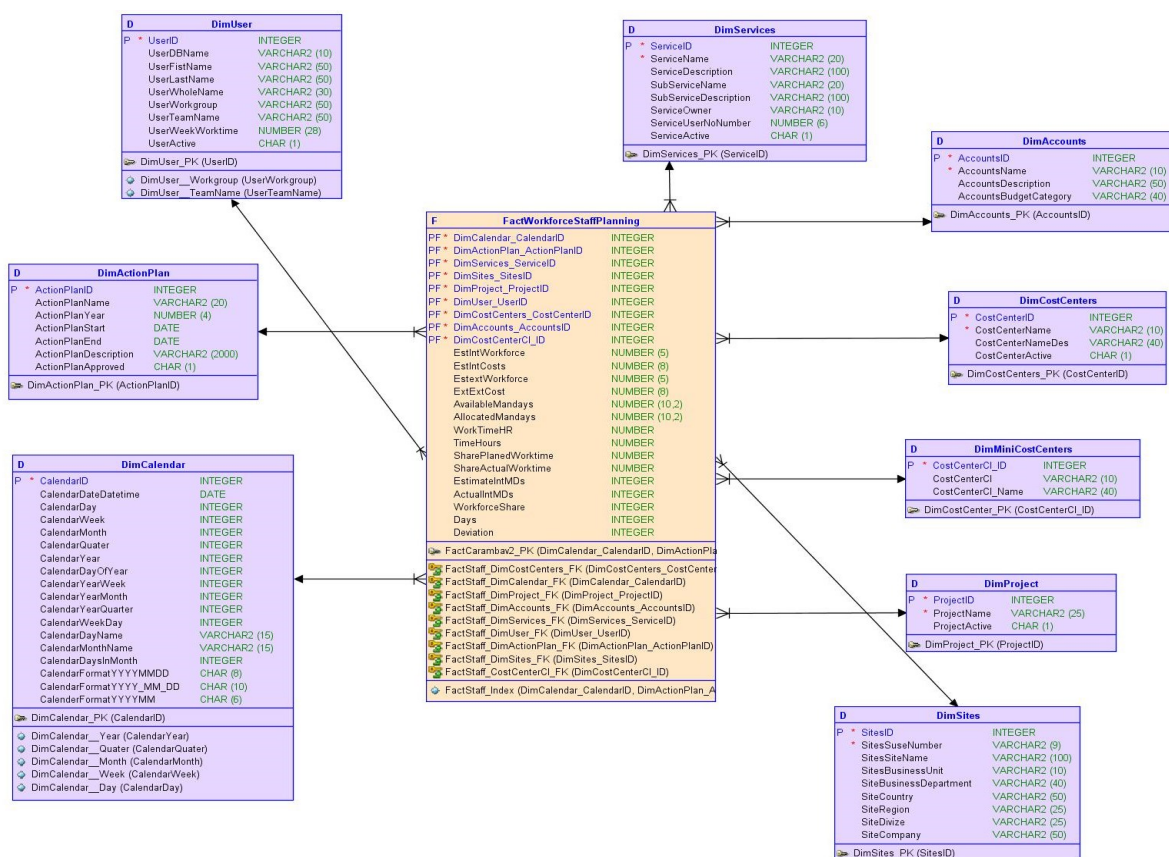
Příloha 4: Celkový konceptuální model



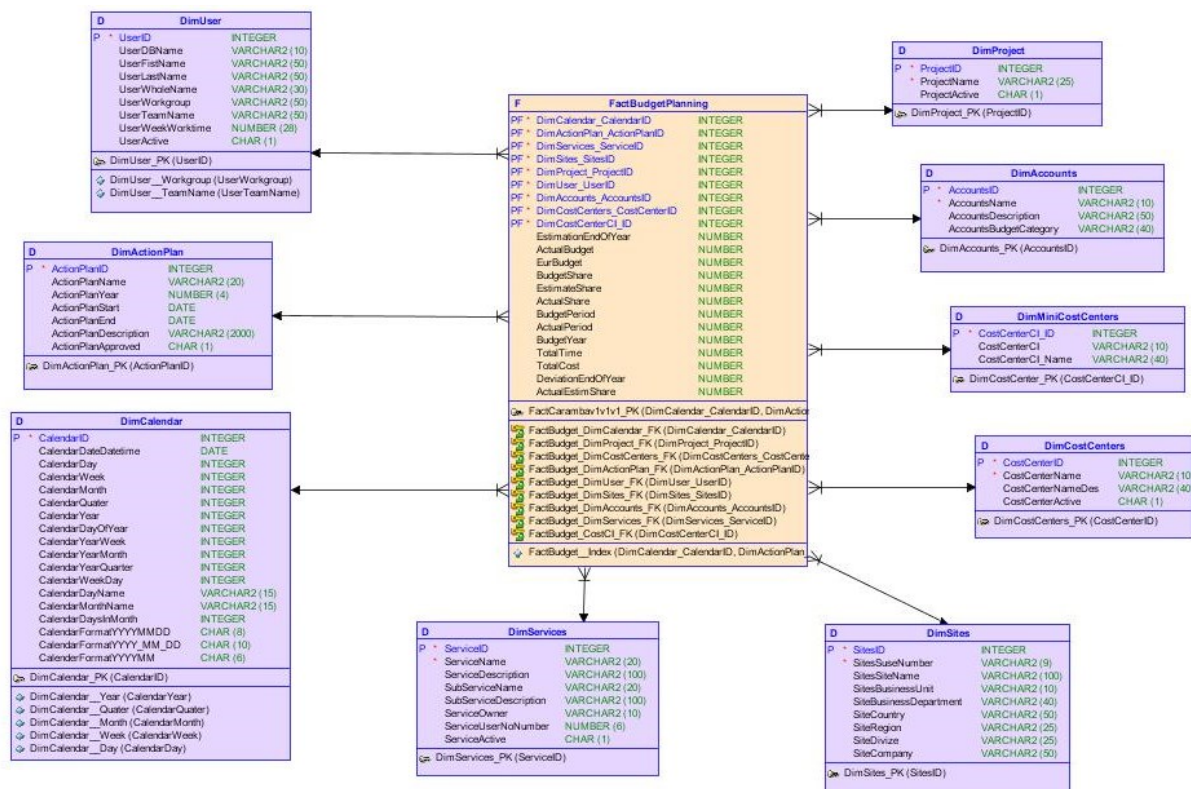
Příloha 5: Celkový logický model



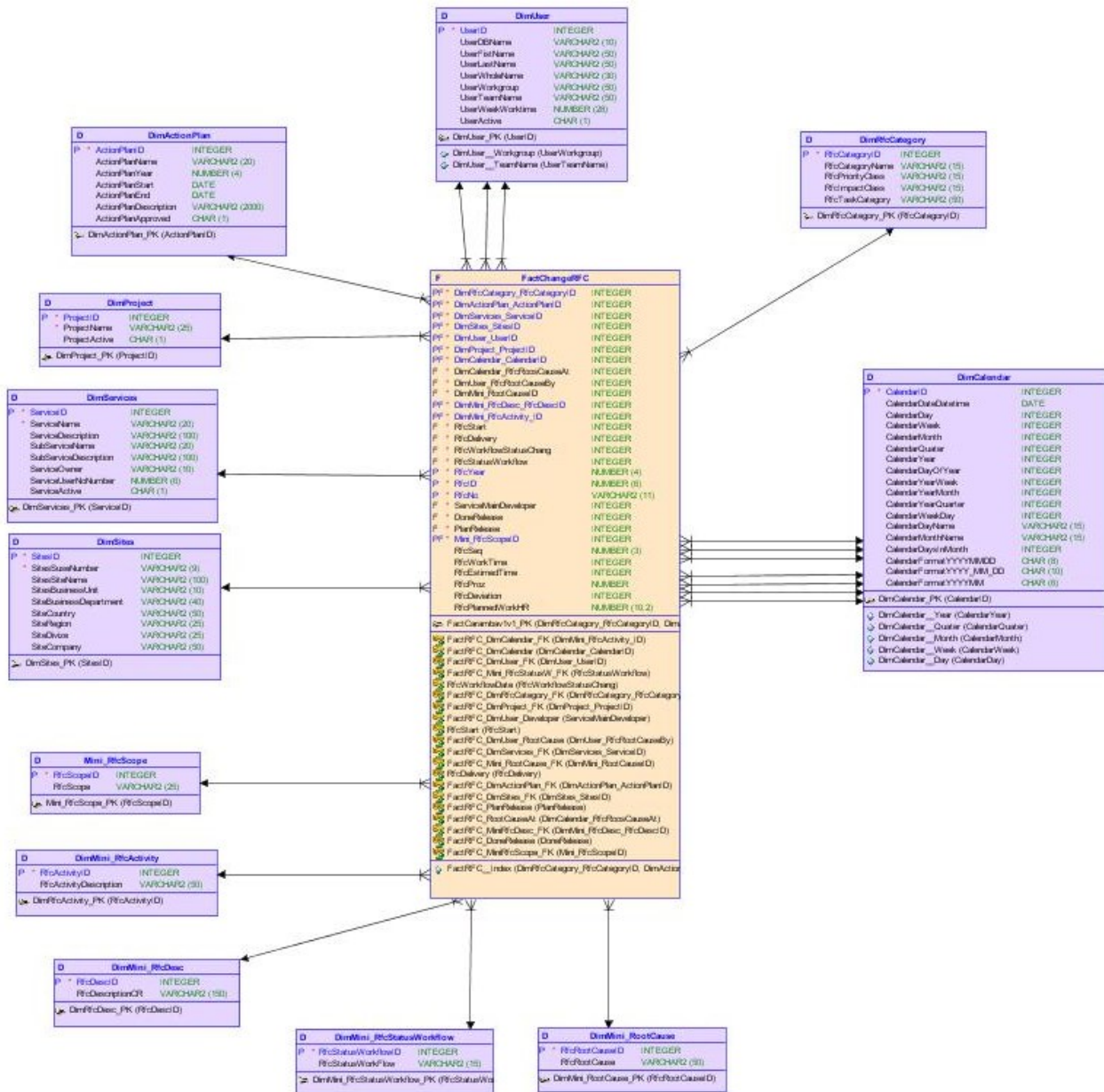
Fyzický datový model pro faktovou tabulku FactWorkflowStaffPlanning



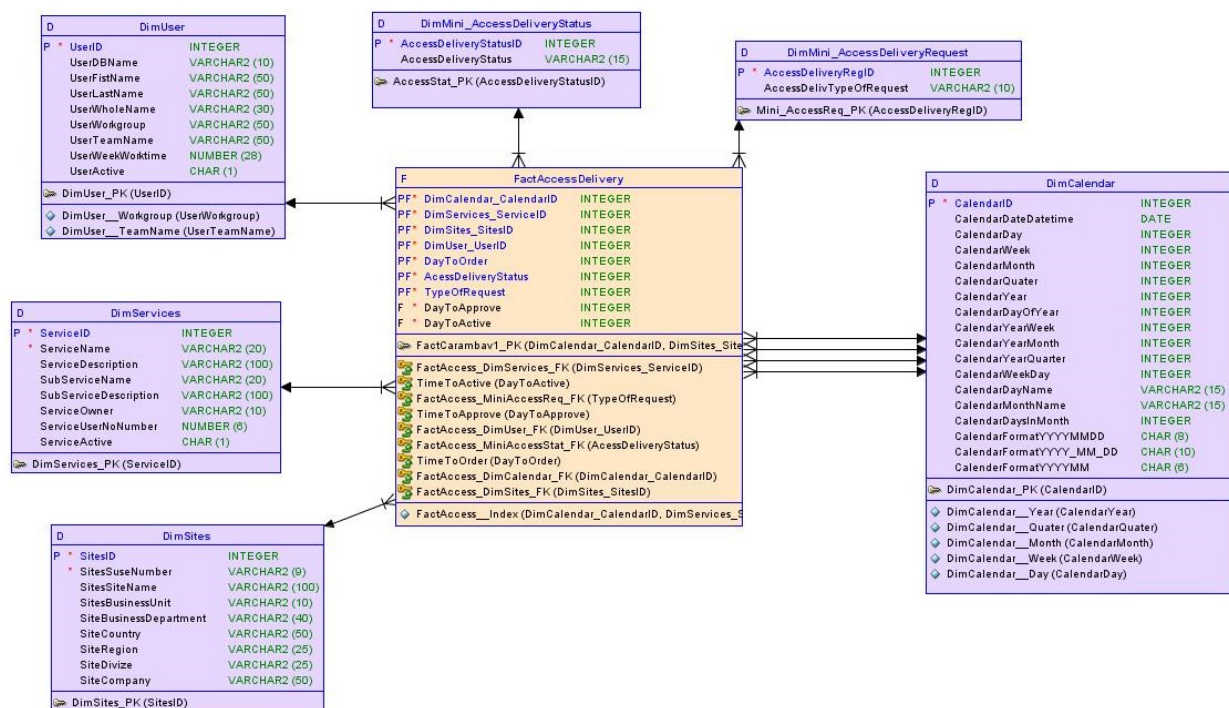
Fyzický datový model pro faktovou tabulku FactBudgetPlanning



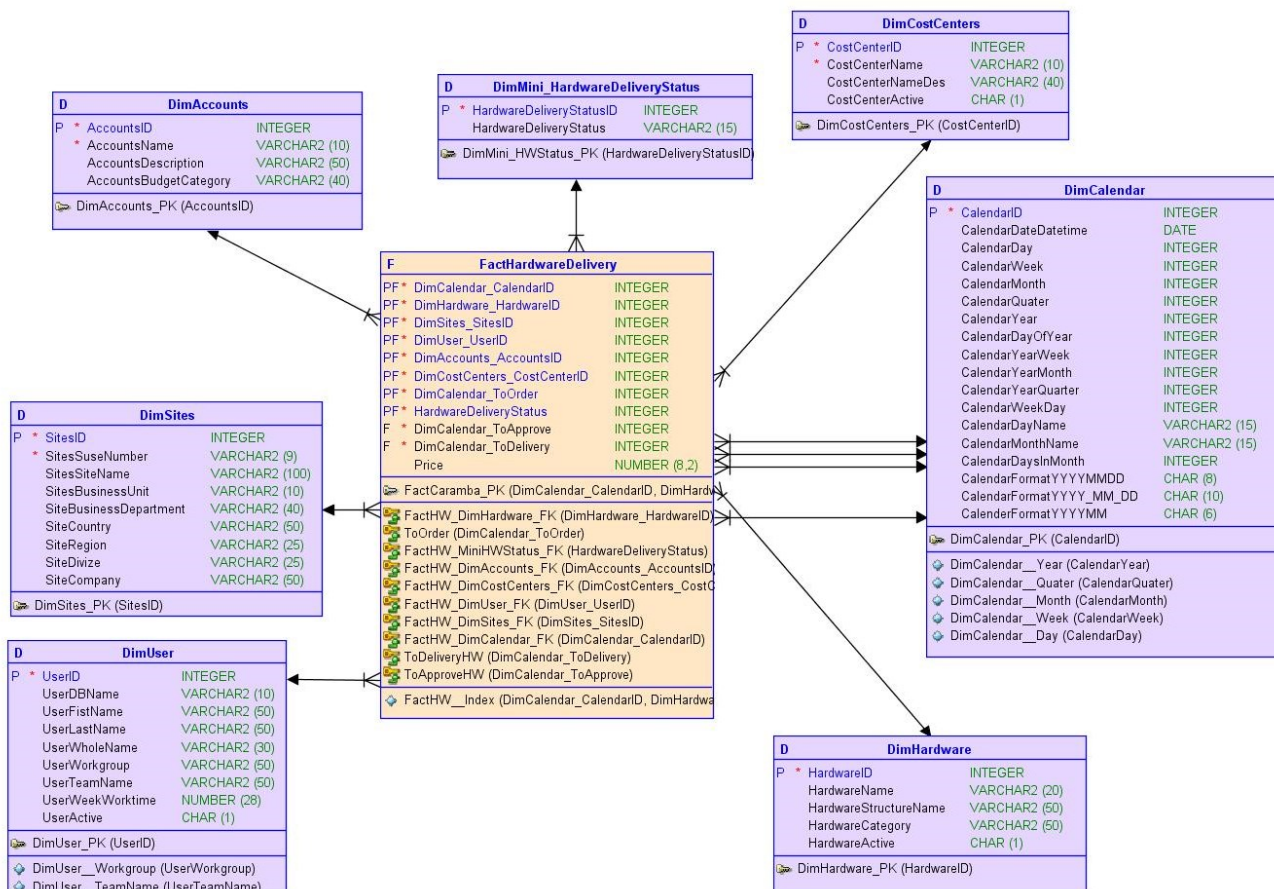
Fyzický datový model pro faktovou tabulku FactChangeRFC



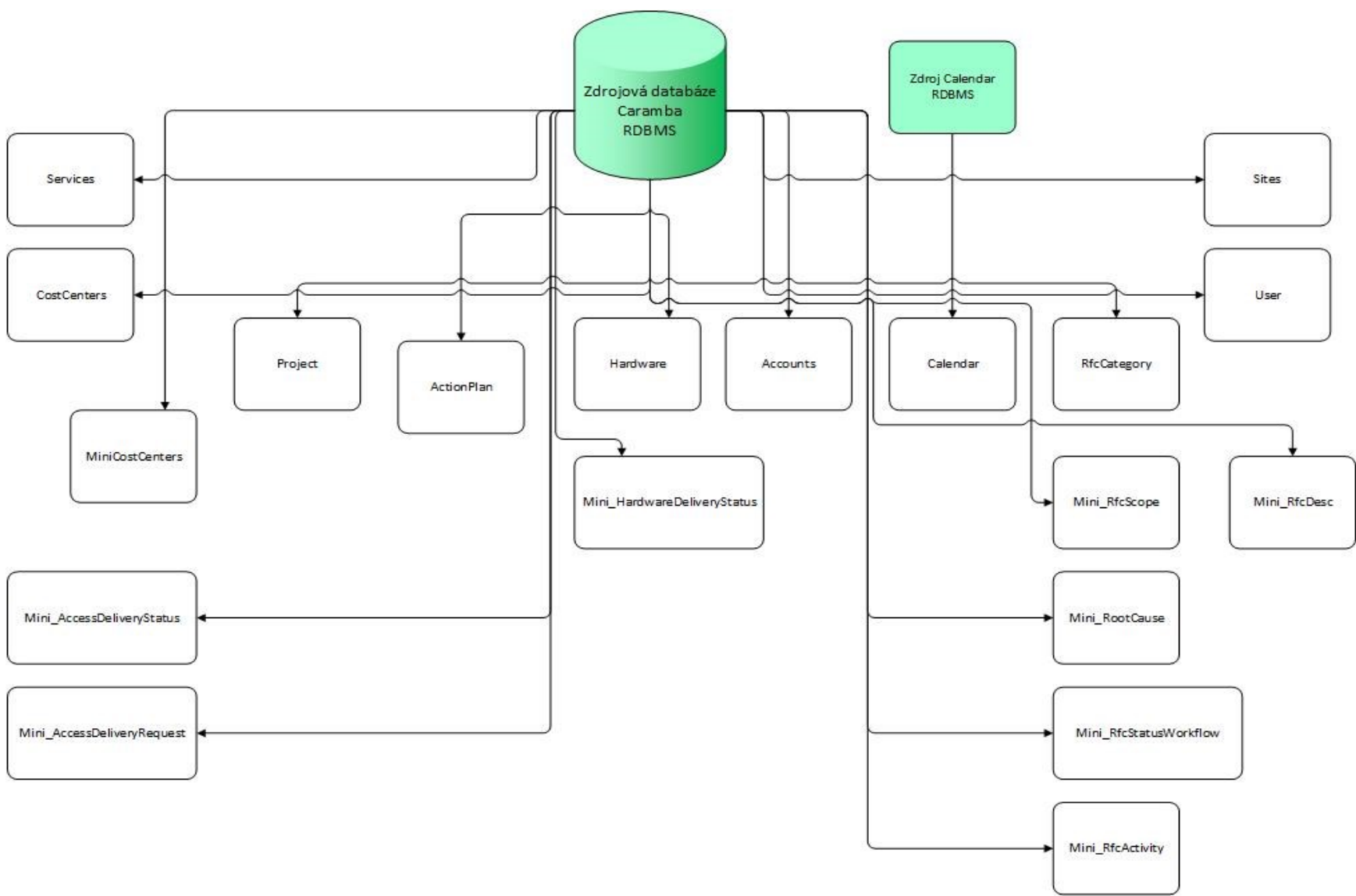
Fyzický datový model pro faktovou tabulku FactAccessDelivery



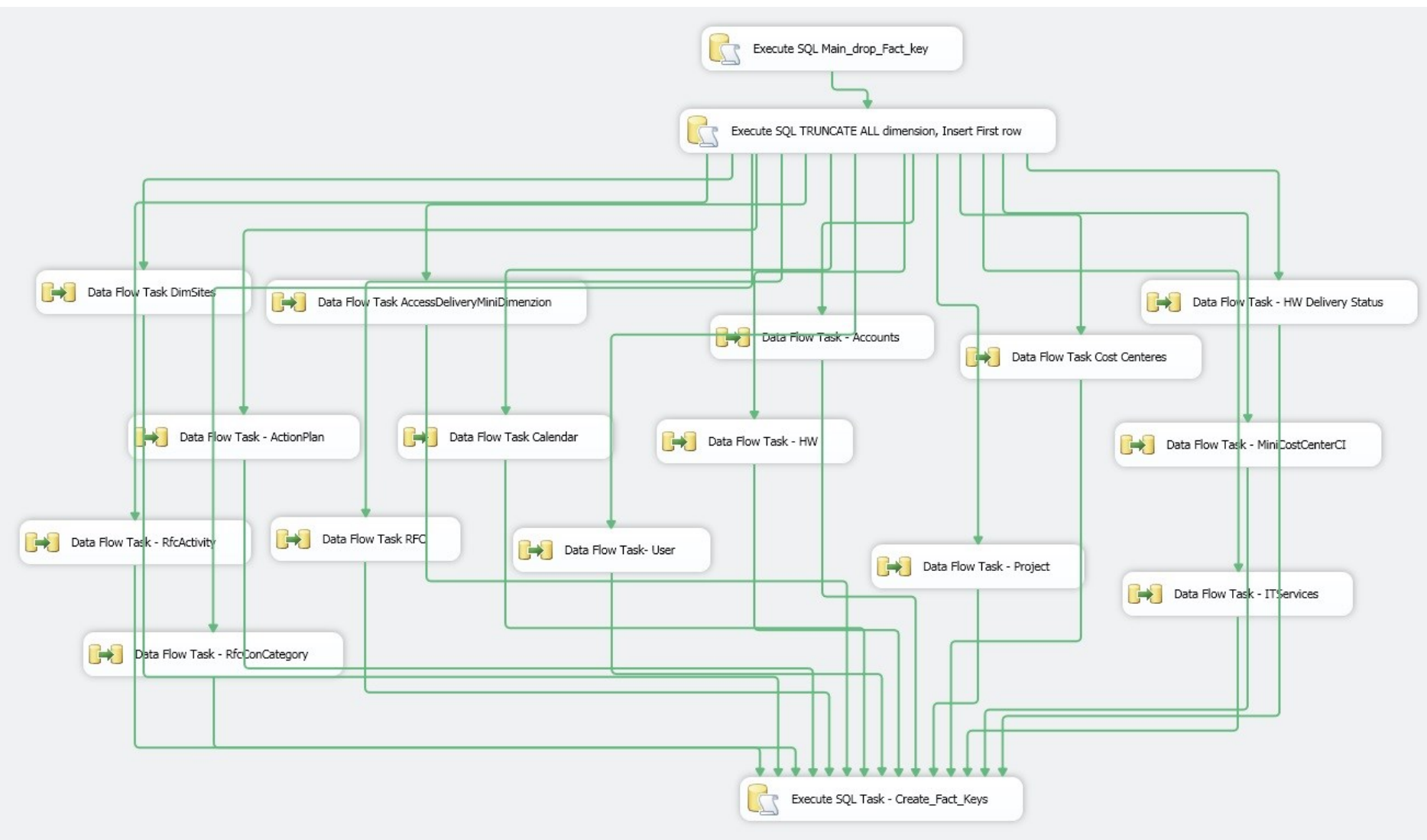
Fyzický datový model pro faktovou tabulku FactHardwareDelivery



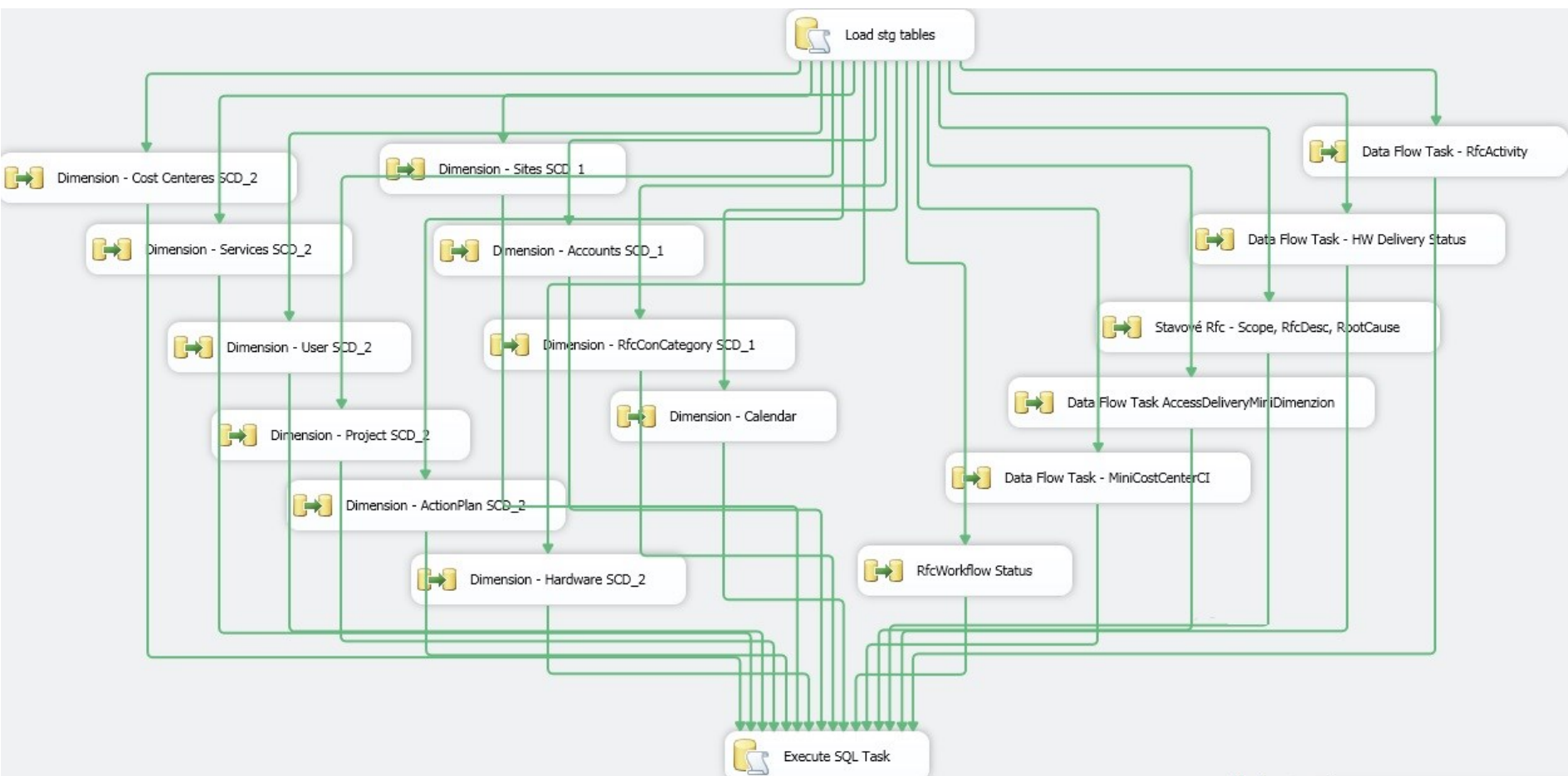
Příloha 7: Vysoko-úrovňový plán (High-level plan)



Příloha 8: Datová pumpa dimenzionálního modelu – historické nahrání



Příloha 9: Datová pumpa dimenzionálního modelu - inkrementální nahrání



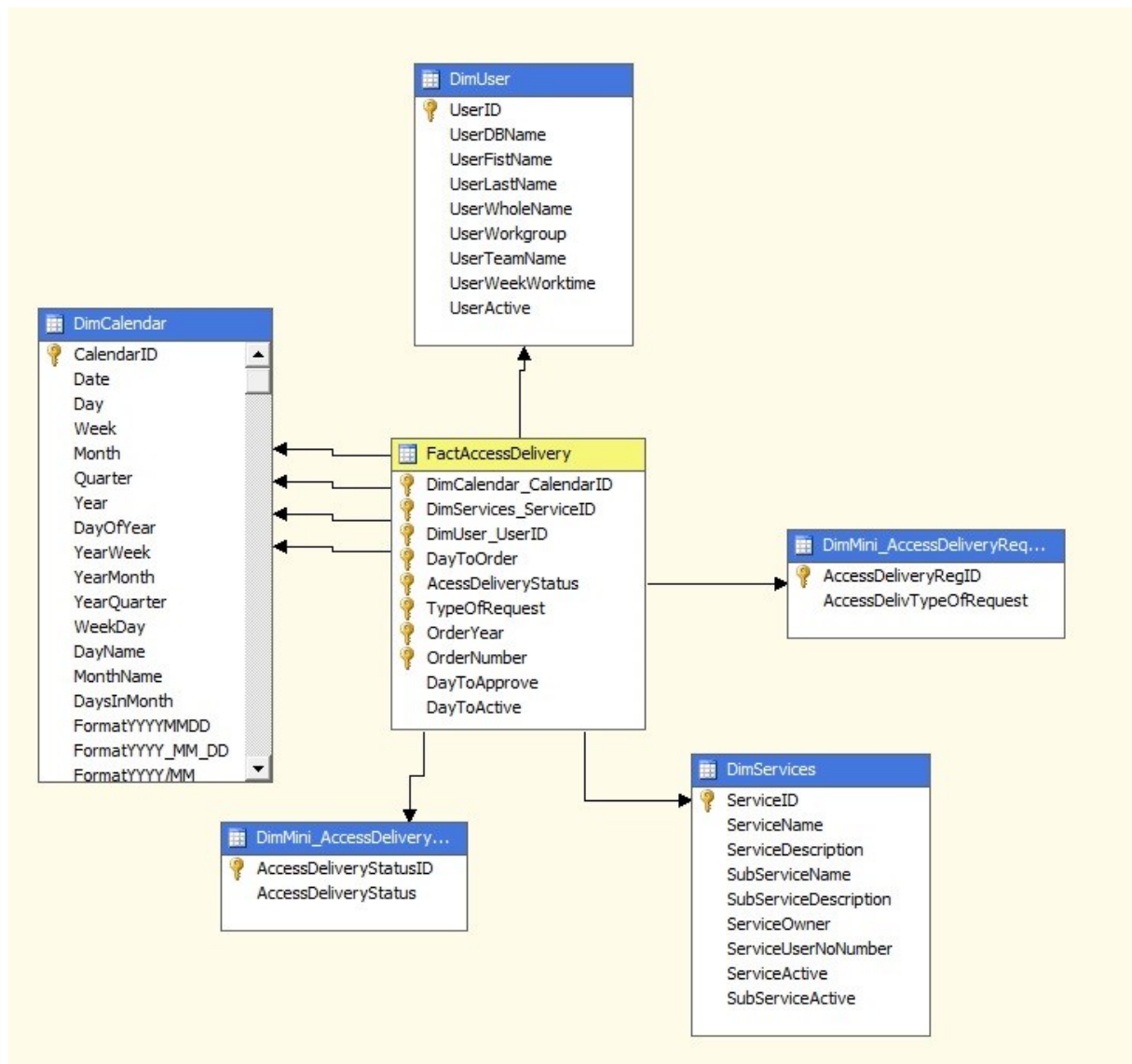
Příloha 10: SQL Skript pro nahrání faktové tabulky FactRfc

```

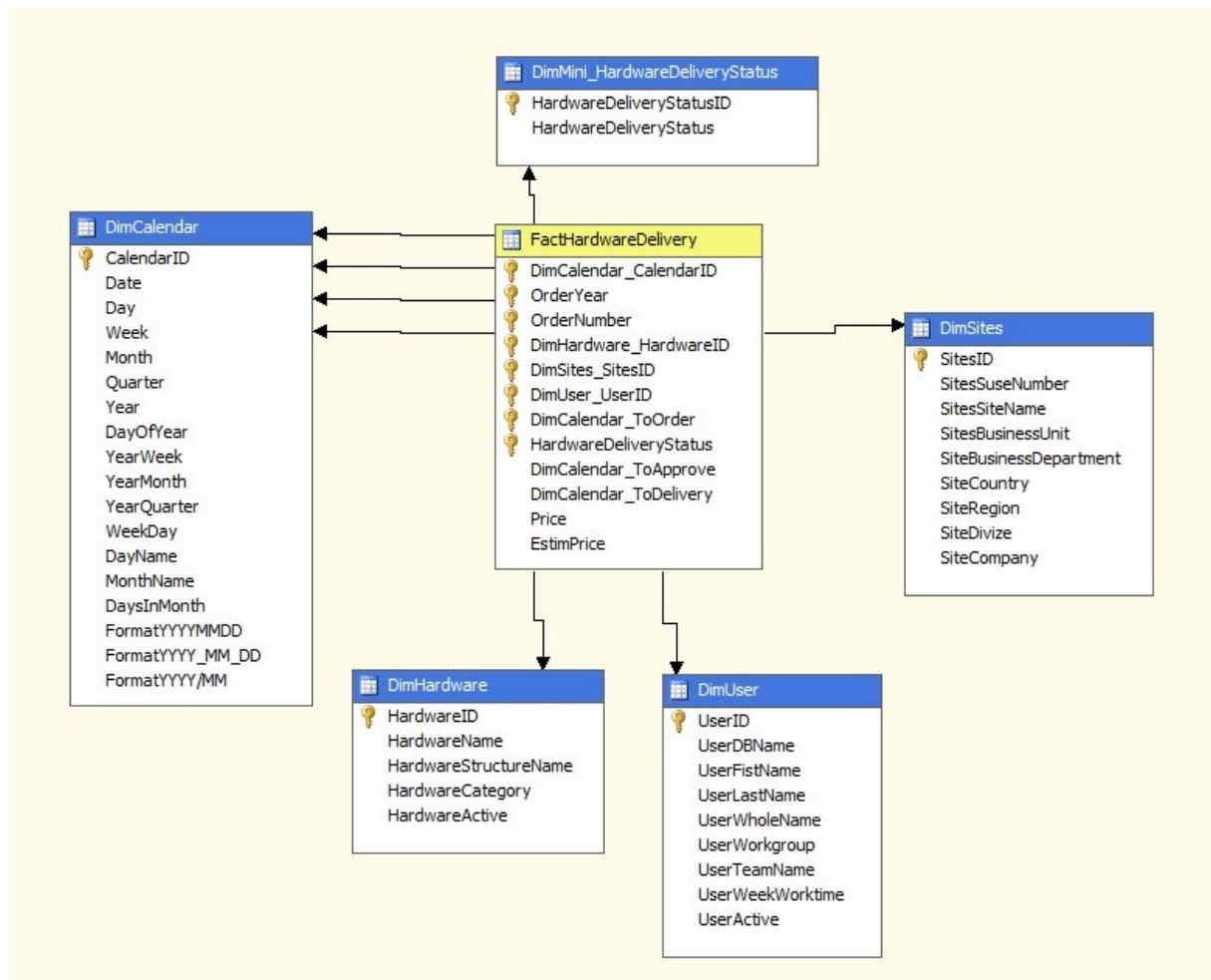
1
2 INSERT INTO [dbo].[FactRfc]
3     ([DimMiniRfcConCategory]
4     , [DimActionPlan_ActionPlanID]
5     , [DimServices_ServiceID]
6     , [DimSites_SitesID]
7     , [DimUser_UserID]
8     , [DimProject_ProjectID]
9     , [DimCalendar_CalendarID]
10    , [DimCalendar_RfcRootCauseAt]
11    , [DimUser_RfcRootCauseBy]
12    , [DimMini_RootCauseID]
13    , [DimMini_RfcDesc_RfcDescID]
14    , [DimMini_RfcActivity_ID]
15    , [RfcStart]
16    , [RfcDelivery]
17    , [RfcWorkflowStatusChang]
18    , [RfcStatusWorkflow]
19    , [RfcYear]
20    , [RfcID]
21    , [RfcNo]
22    , [ServiceMainDeveloper]
23    , [DoneRelease]
24    , [PlanRelease]
25    , [Mini_RfcScopeID]
26    , [DimMiniPriorityImpact]
27    , [RfcSeq]
28    , [RfcWorkTime]
29    , [RfcEstimatedTime]
30    , [RfcProz]
31    , [RfcDeviation]
32    , [RfcPlannedWorkHR])
33
34 SELECT
35     DimMiniRfcConCategory = ISNULL(A.RfcConCategoryID, 0)
36     , DimActionPlan_ActionPlanID = ISNULL(B.ActionPlanID, 0)
37     , DimServices_ServiceID = ISNULL(C.ServiceID, 0)
38     , DimSites_SitesID = ISNULL(D.SitesID, 0)
39     , DimUser_UserID = ISNULL(E.UserID, 0)
40     , DimProject_ProjectID = ISNULL(F.ProjectID, 0)
41     , DimCalendar_CalendarID = ISNULL(G.CalendarID, 0)
42     , DimCalendar_RfcRootCauseAt = ISNULL(CalA.CalendarID, 0)
43     , DimUser_RfcRootCauseBy = ISNULL(CByUser.RfcRootCauseBy, 0)
44     , DimMini_RootCauseID = ISNULL(RCM.RootCauseID, 0)
45     , DimMini_RfcDesc_RfcDescID = ISNULL(L.RfcDescID, 0)
46     , DimMini_RfcActivity_ID = ISNULL(Q.RfcActivity_ID, 0)
47     , RfcStart = ISNULL(CalS.CalendarID, 0)
48     , RfcDelivery = ISNULL(CalDEL.CalendarID, 0)
49     , RfcWorkflowStatusChang = ISNULL(CalCh.CalendarID, 0)
50     , RfcStatusWorkflow = ISNULL(T.StatusWorkflowID, 0)
51     , RfcYear = ISNULL(X.RfcYear, '0000')
52     , RfcID = ISNULL(X.RfcID, '0')
53     , RfcNo = ISNULL(X.RfcNo, 'Unknown')
54     , ServiceMainDeveloper = ISNULL(SMainDev.ServiceMainDeveloperID, 0)
55     , DoneRelease = ISNULL(CalD.CalendarID, 0)
56     , PlanRelease = ISNULL(CalP.CalendarID, 0)
57     , Mini_RfcScopeID = ISNULL(P.RfcScopeID, 0)
58     , RfcSeq = X.PRIORITYSEQUENZED
59     , RfcWorkTime = R.RCONRESOURCEQUANT
60     , RfcEstimatedTime = ConAC.ESTIMTIME
61     , RfcProz = ConAC.PROZHOURS
62     , RfcDeviation = ISNULL((((CAST(CAST(AB.ESTIMATE AS float) AS numeric(8,2)))) - (CAST(CAST(AB.actualcosts AS float) AS numeric(8,2))))), 0)
63     , RfcPlannedWorkHR = ConAC.ACTUALW
64     , DimMiniPriorityImpact = ISNULL(MI.PriorityImpactID, 0)
65
66 FROM Rfcxx X
67 LEFT JOIN stg.ITRfcChanges R ON X.RFCYEAR = R.RFCYEAR AND X.RFCID = R.RFCID AND X.APPID = R.APPID
68 LEFT JOIN dbo.DimCalendar C ON CONVERT(CHAR(10), X.CREATEDAT) = C.FormatYYYY_MM_DD
69 LEFT JOIN dbo.DimCalendar CalD ON CONVERT(CHAR(10), X.DONERELEASE) = CalD.FormatYYYY_MM_DD
70 LEFT JOIN dbo.DimCalendar CalP ON CONVERT(CHAR(10), X.PLANNEDRELEASE) = CalP.FormatYYYY_MM_DD
71 LEFT JOIN dbo.DimCalendar CalA ON CONVERT(CHAR(10), X.ROOTCAUSEAT) = CalA.FormatYYYY_MM_DD
72 LEFT JOIN dbo.DimCalendar CalS ON CONVERT(CHAR(10), X.STARTDATE) = CalS.FormatYYYY_MM_DD
73 LEFT JOIN dbo.DimCalendar CalDEL ON CONVERT(CHAR(10), X.DELIVERYDATE) = CalDEL.FormatYYYY_MM_DD
74 LEFT JOIN dbo.DimCalendar CalCh ON CONVERT(CHAR(10), X.CHANGED) = CalCh.FormatYYYY_MM_DD
75 LEFT JOIN dbo.DimUser CByUser ON X.ROOTCAUSEBY = CByUser.UserID
76 LEFT JOIN dbo.DimMini_RootCause RCM ON X.ROOTCAUSEID = RCM.RfcRootCause
77 LEFT JOIN dbo.DimMini_RfcDesc L ON X.RFCNAME = L.RfcDescriptionCR
78 LEFT JOIN dbo.DimMini_RfcStatusWorkflow T ON X.STATUSWORKFLOW = T.RfcStatusWorkflow
79 LEFT JOIN dbo.DimMini_RfcScope P ON X.SCOPE = P.RfcScope
80 LEFT JOIN dbo.DimMiniPriorityImpact MI ON X.PRIORITYCLASSID = MI.PriorityClass
81 AND X.IMPACTCLASSID = MI.ImpactClass
82 LEFT JOIN dbo.DimMini_RfcActivity Q ON X.RFCACTIVITY = Q.RfcActivityDescription
83 LEFT JOIN stg.ITRfcConCategory ConCA ON X.RCONCATEGORY = ConCA.RCONCATEGORY
84 LEFT JOIN dbo.DimMiniRfcConCategory A ON X.RCONCATEGORY = A.RfcCategoryName
85 LEFT JOIN stg.ITServices StgSer ON CONVERT(varchar(20), X.APPID) = StgSer.SERVICEID
86 LEFT JOIN dbo.DimServices C ON StgSer.SERVICEID = C.ServiceName
87 LEFT JOIN stg.ITActionPlan StgActPlan ON CAST(X.APPID AS VARCHAR(20)) = StgActPlan.SERVICEID
88 AND X.ACTIONPLANID = StgActPlan.ACTIONPLANID AND X.BUDGETYEAR = StgActPlan.[BUDGETYEAR]
89 LEFT JOIN dbo.DimActionPlan B ON StgActPlan.ACTIONPLANID = B.ActionPlanName
90 AND StgActPlan.[BUDGETYEAR] = B.[ActionPlanYear] AND StgActPlan.SERVICEID = B.[ServiceName]
91 LEFT JOIN dbo.DimSites D ON X.Number = D.NumberID
92 LEFT JOIN dbo.DimUser SMainDev ON X.RESPERSON = SMainDev.UserID
93 LEFT JOIN dbo.DimUser E ON X.DBUSER = E.UserID
94 LEFT JOIN dbo.DimProject F ON X.Project = F.[ProjectName]
95 LEFT JOIN dbo.ViewX AB ON X.RFCYEAR = AB.RFCYEAR AND X.RFCID = AB.RFCID
96 LEFT JOIN stg.ITRfcConsActivity ConAC ON X.rfcyear = ConAC.rfcyear AND X.[RFCID] = ConAC.rfcid
97
98 GO
99 --

```

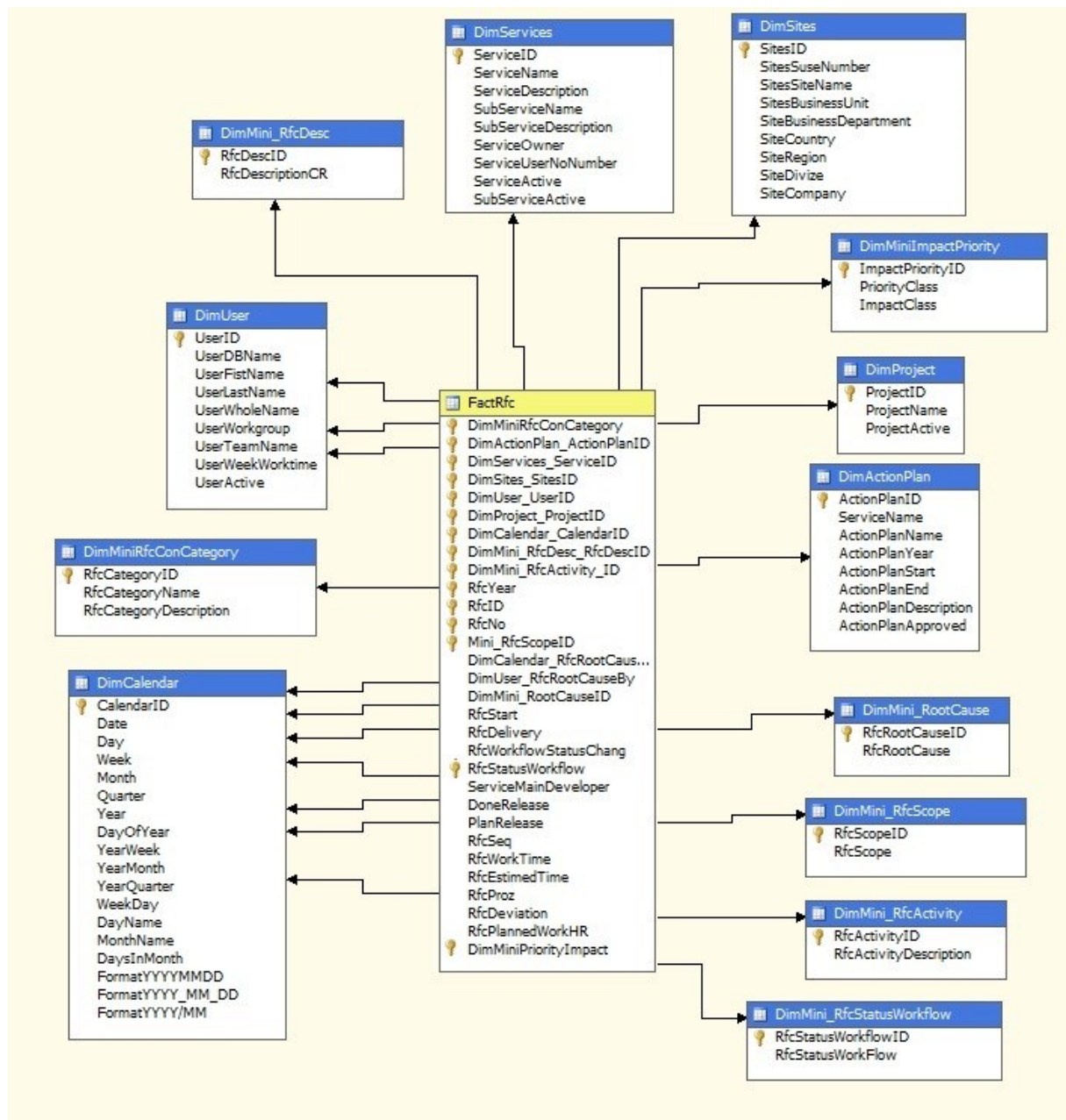
Příloha 11: OLAP kostka pro faktovou tabulku FactAccessDelivery



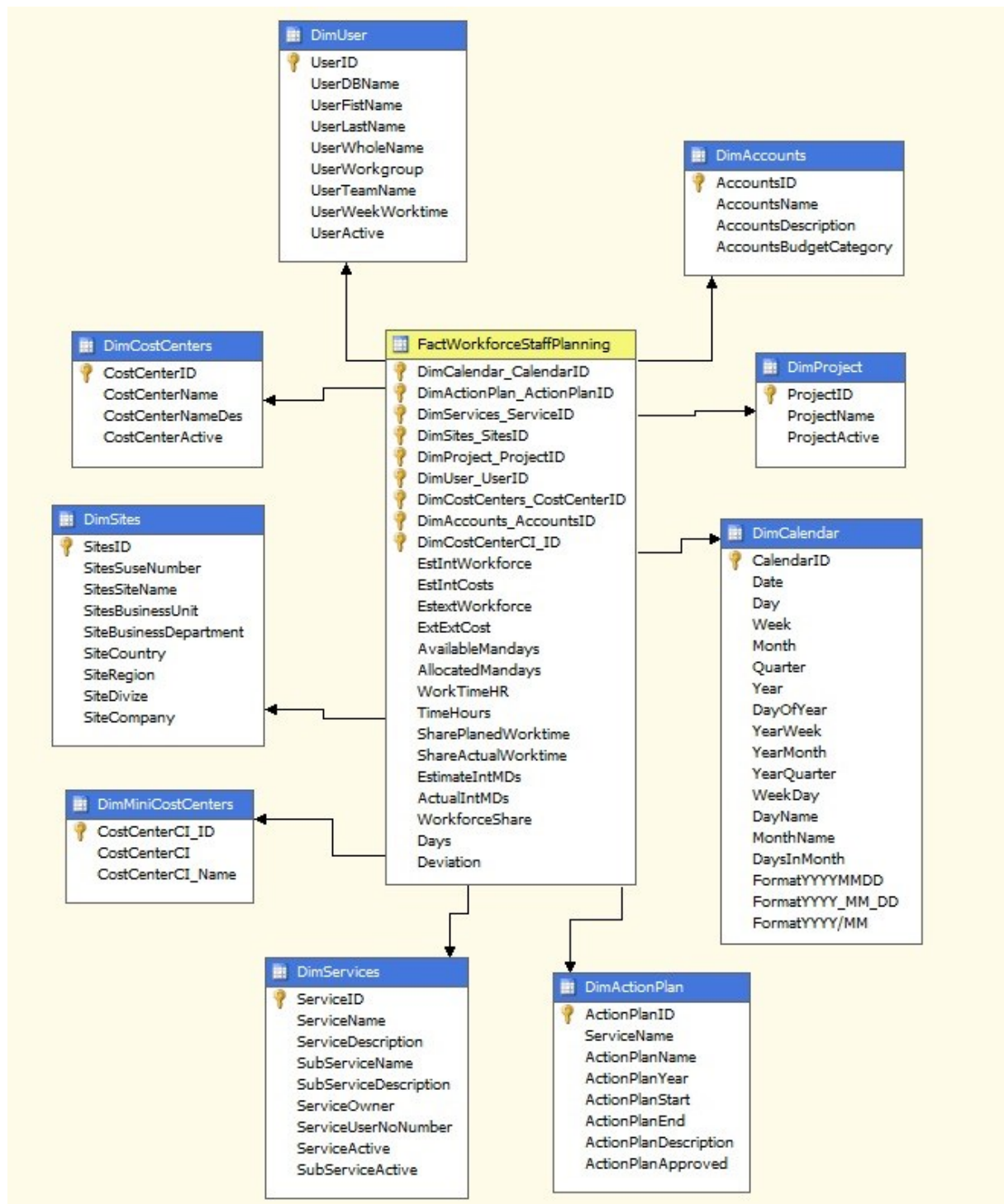
Příloha 12: OLAP kostka pro faktovou tabulku FactHardwareDelivery



Příloha 13: OLAP kostka pro faktovou tabulku FactRfc



Příloha 14: OLAP kostka pro faktovou tabulku FactWorkforceStaffPlanning



Příloha 15: OLAP kostka pro faktovou tabulku FactBudgetPlanning

